ESD-TDR-64-395

ESPOD MATHEMATICAL AND SUBROUTINE DESCRIPTION

TECHNICAL DOCUMENTARY REPORT NO. ESD-TDR-64-395

JUNE 24, 1964

496L Systems Program Office
ELECTRONIC SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
L. G. Hanscom Field, Bedford, Massachusetts

PROJECT ES-3-496L-3627

Prepared under Contract AF 19(628)-594

STL NO. 8497-6065-RU-000

ESD-TDR-64-395

ESPOD Mathematical and Subroutine Description

Technical Documentary Report No. ESD-TRD-64-395

June 24, 1964

496L Systems Program Office
Electronic Systems Division
Air Force Systems Command

United States Air Force

L. G. Hanscom Field, Bedford, Massachusetts

Project ES-3-496L-3627

Prepared Under Contract AF 19(628)-594

STL No. 8497-6065-RU000

FOREWORD

This document is one of three reports which describe ESPOD, a general satellite orbit determination program prepared for the Air Force Electronics System Division for use in the Spacetrack/SPADATS Center at Ent Air Force Base, Colorado Springs, Colorado.

This report is

ESD-TDR-64-395	ESPOD Mathematical and Subroutine Description (STL No. 8497-6065-RU000)

The companion reports are

ESD-TDR-64-393	ESPOD Functional Description (STL No. 8497-6067-RU000)
ESD-TDR-64-394	ESPOD Operating Instructions and Card Formats (STL No. 8497-6066-RU000)

The ESPOD program was prepared by TRW Space Technology Laboratories under Air Force Contract Number AF 19(628)-594.

CONTENTS

CONTENTS (Continued)

ILLUSTRATIONS

ILLUSTRATIONS (Continued)

TABLES

1. INTRODUCTION

The primary function of this report is to provide the specialist in the field an in-depth treatment of the mathematical and related computer processes used to develop the ESPOD program. Secondarily, it is arranged to include the fundamental organization of the program, the background and general information relative to its options and the computer subroutines and coordinate systems.

For those interested in an abridged technical discussion of ESPOD it is recommended that ESD-TDR-64-393, ESPOD Functional Description, be consulted.

The analyst-operator requiring operational information should consult ESD-TDR-64-394, ESPOD Operating Instructions and Card Formats.

2. ESPOD FUNCTIONAL FLOW

The following description gives the order of functions and control of the more important options. A summary block diagram is shown in Figure 2-1; the Guide Flow Charts (see Section 2.3.1.1, Section 2.3.2.1 and Section 2.3.3.1) show more detail.

2.1 ESPOD FUNCTIONAL OPERATIONS

The entire ESPOD program is divided into three main segments: a data preprocessor, ESPØD; a differential corrector, ESPØDDC; and an ephemeris generator, ESPØDEPH.

2.1.1 Preprocessor Operations

The preprocessor (ESPØD) processes all the input data and acts as the main control package for the various program options available. It interprets the program control data input, the satellite observations input, and the other program modules. It controls the calling of the differential corrector (ESPØDDC) and the ephemeris generator (ESPØDEPH) and their subsequent operations.

The flow of the logic through ESPØD is controlled by the flags set on the Job Description Card (JDC), the first, unique, and mandatory card for each separate case in the input deck. Following the JDC in the deck setup are the preliminary data cards (specifying the initial conditions, the variables to be solved, the options, constant changes, etc.), the observation cards (unless observations are read from the SRADU tape), and sensor cards (if file sensor data must be augmented or changed).

Column 30 of JDC card is used to indicate whether the current run is starting from a "cold start" or a "conditioned start." The distinction is that on a conditioned start the observations and sensor data will be recovered from a tape which was generated on a previous run of the same case. This eliminates the necessity of reprocessing observation cards when the only change in an input deck is in the JDC card and in the preliminary data.

The following is the procedure for input processing from a cold start: the JDC card is read and loaded into core, followed by the preliminary data cards which are interpreted and organized into core storage. At this point, the program interrogates the JDC card to see if this is to be a postprocessor run only. If so, ESPØDEPH is called into core from the B2 master tape and control is transferred to it. If ESPØDDC is to be executed, the observation cards, it any, are read and time sorted. Sensor cards, if any, are then processed, and a tape is generated of the input data. ESPØDDC is called from the B2 tape and control given to it. After execution of ESPØDDC, the JDC card is again interrogated and ESPØDEPH is called and executed if requested.

2.1.2 Differential Corrector Operations

The function of the differential correction package (ESPØDDC) is to determine the orbit which best fits, in the least squares sense, the observational data and as a result to improve the initial values of the solution variables. To perform this function, ESPØDDC interfaces with the preprocessor (ESPØD) and the postprocessor (ESPØDEPH), and the observations, which have been formatted, assigned weights, time sequenced, and written on an observation tape. Starting with an initial estimate of the position and velocity of the vehicle at some specified epoch time, a simulated trajectory is calculated and the "observed" minus "computed" residuals are formed. The overdetermined system of linear equations generated is solved, in a least squares sense, for the differential corrections to the solution variables.

ESPØDDC is composed of three major subpackages; the trajectory package, the radar partials package, and the least squares package.

2.1.2.1 Trajectory Package (TRAJ)

The orbit is simulated through the integration of the equations of motion using Cowell's scheme, with the inclusion of the various perturbing forces controlled by input flags. The variational equations of α_o, δ_o, β_o, A_o, r_o, v_o, drag and $C_D A/2m$ and drag variation K also are integrated as needed.

The times at which the trajectory package provides outputs are selected one at a time from the tape of the observations which was generated in the

preprocessor. All times before the epoch are handled first, in chronologically reverse order from epoch, to obviate restarting the integration more than twice on any one iteration.

2.1.2.2 Radar Partials Package (RADR)

At each observation time, control is passed from the trajectory package to the radar partials package. This package sets up the partial derivatives of the observation (range, azimuth, elevation, range rate, hour angle, and declination) with respect to each parameter in the solution vector. The observed minus computed residual is formed and an overdetermined system of equations AX = B is built up in a form suitable to the least squares package.

The radar partials package outputs the station number and name, observation time (minutes from epoch), numbered residual (tagged with an asterisk, if deleted), and a summary table of the mean and RMS of the residuals for each station.

2.1.2.3 Least Squares Package (LEGS2)

The least squares package processes the matrix of partial derivatives and residuals set up by the radar partials package in order to improve the solution variables. Bounds are imposed on the size of the corrections made to each variable in order to keep the differential corrections sufficiently small. On each pass through this package, four sets of differential corrections are calculated corresponding to the current values of bounds, bounds/2, bounds/4, and bounds/8. The latter three are contingent solutions supplied only if the nominal fails to produce an improvement in the total RMS of the residuals. On an iteration which is converging as predicted, the bounds may be doubled to permit faster ultimate convergence. Convergence is assumed complete when the predicted value of the RMS of the residuals is within a specified percentage of the current value.

The least squares package outputs the differential correction to be applied to each variable, the old value of the variable, the new value, the uncertainty, the bound, the current RMS, predicted RMS, the best RMS so far, the covariance matrix, the correlation matrix, and a message to indicate whether the solution is converging.

2.1.3 Ephemeris Predictor

The ephemeris predictor (ESPØDEPH) calculates the position and velocity of the satellite for a set of times specified in the preliminary data. It interfaces with the preprocessor, the differential corrector, the on-line output printer, and an output parameter tape. Options are available (under the control of the JDC card) for updating an input covariance matrix, punching an a priori S matrix, and generating a binary ephemeris tape in SPADATS format. ESPØDEPH receives its input from a tape which has been generated either in ESPØDDC or ESPØD.

2.1.3.1 Trajectory Package

Starting with the initial position and velocity of the vehicle at some epoch, the trajectory is simulated using a trajectory package identical to the one in ESPØDDC. However, output will occur every Δt minutes up to T minutes from epoch or at the times specified by the DAC card or at the times specified by the PRDCT cards. The trajectory output block consists of the osculating elements reflecting all the perturbations present in the integration model. Also included are the vehicle position and velocity in polar and Cartesian form, the classical orbital elements, vehicle altitude, latitude (geodetic), longitude, argument of latitude, period, perigee and apogee altitudes, time from epoch to the next crossing of the ascending node, and a set of indeterminacy free elements. Regardless of the options requested, this output block is printed. If requested, a binary ephemeris tape containing the Cartesian position and velocity of the vehicle versus time will be generated.

2.1.3.2 Matrix Update Package (UPDAT)

The matrix update package will update a 6 x 6, 7 x 7 or 8 x 8 covariance matrix to the times specified in the Δt, T list. The six elements of the vehicle position and velocity, the uncertainties in drag, and the uncertainties in variation in drag may be supplied for update. The package rotates the updated covariance matrix to the orbit plane (U, V, W), Cartesian, and polar equivalents, and outputs the uncertainties and correlations in the elements of each type. These are labeled as sigma and rho matrices.

The option also exists (by a flag on the JDC card) for inserting the updated covariance matrix to punch an a priori A^TA matrix on cards in a format suitable for input to the ESPØDDC.

2.2 EXPLANATION FOR SUBROUTINE FLOW

Figure 2-1 is a summary block diagram illustrating the segments of the program, an outline of their functions, and the flow among them.

2.2.1 Description of Subroutine Flow

The subroutine flow given for ESPOD presentation departs from the conventional block diagram description of a program in that it names in order each subroutine called by the program. Principal decisions in the process are also indicated and important calculations are noted. Subroutines which are called by a previous subroutine and which operate under its control are listed indented one column to its right. Thus the greatest detail, indented many times, appears most often at the right side of the page, while the main flow is outlined in the first three or four columns on the left side of the page. Minor and utility subroutines which are used in many places [for example, ASIN (arc sine)] have been omitted.

2.2.2 Aids to Using Subroutine Flow

Two aids are provided for easier interpretation of the subroutine flow. The first is the guide flow chart which summarizes by functional blocks the separate detailed processes. Each block of the guide flow chart indicates the page(s) and lines of the subroutine flow which accomplish its function. The guide flow precedes the detailed flow. The second is a subroutine listing which tells what each subroutine does. The listing is complete for each section of the detailed flow and follows immediately after it. For more detail concerning the subroutine, see Section 4.

2.2.3 Application of Subroutine Flow

The subroutine flow charts are given in three sections corresponding respectively to the three segments of the program ESPØD (Section 2.3.1.2), ESPØDDC (Section 2.3.2.2), and ESPØDEPH (Section 2.3.3.2). The subroutine flow charts have the following uses:

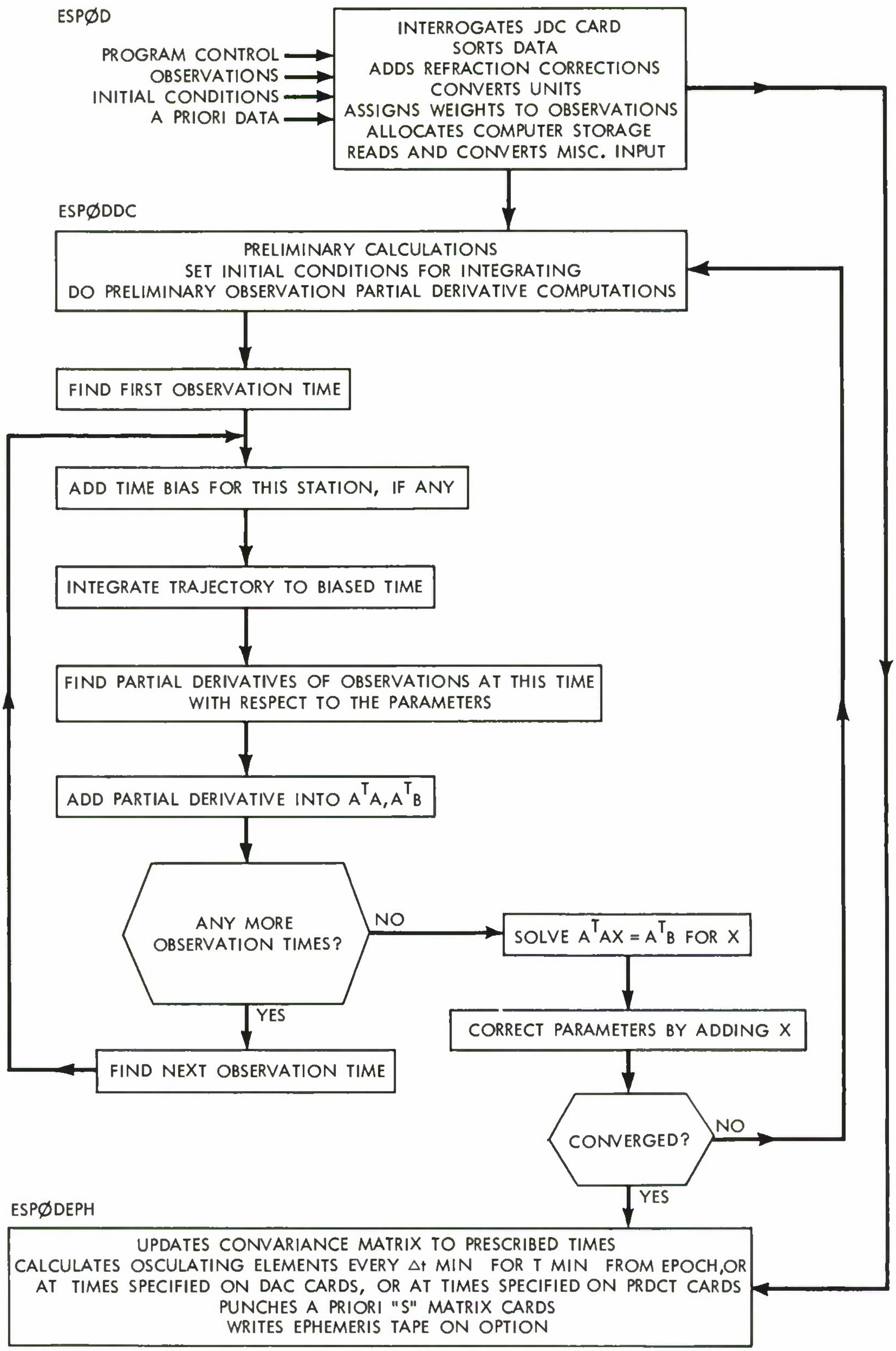


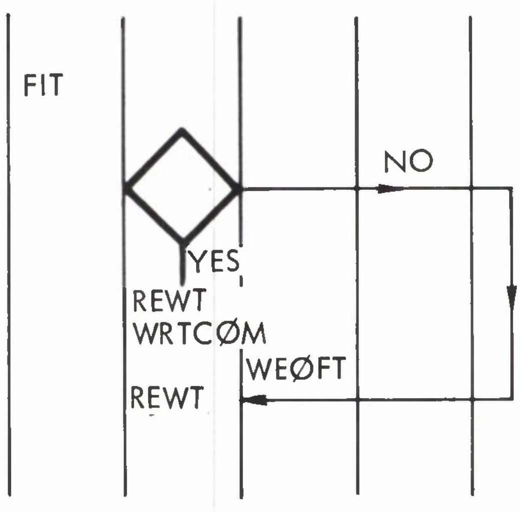
Figure 2-1. Summary Block Diagram

- They show the sequence in which subroutines are used
- They are a master flow chart of the detailed program functions
- They are an index to the subroutines
- They are a guide to the code edit
- They show the interdependence of the subroutines
- They show parallel structure in the different segments.

2.2.4 Interpretation of Subroutine Flow

Each subroutine listed in the flow charts is indented under the subroutine which was responsible for its being called into the program. To aid in recognizing the rank of each subroutine in the heirarchy, vertical lines serve as indentation markers. Subroutines of equal rank are indented equal amounts.

The rank of decisions made within the program will be defined by the indentation line which is nearest the left side of the symbol representing the decision. For example,



FIT calls first for a decision; if the decision is yes, then FIT calls REWT; then WRTCOM which calls WEOFT; FIT then calls REWT. (If decision is no, the flow bypasses to REWT, which is called by FIT.) The decision box has the same rank as REWT and WRTCOM, and they are all subservient to FIT.

Figure 2-5,(lines 16 through 22, page 2-27), the cycling of BOUNDS and LEGS2 may be confusing. In this section, LEGS2 is forming the solution for the different scaling of BOUNDS, that is, the nominal BOUNDS, BOUNDS/2, BOUNDS/4, BOUNDS/8. The first time LEGS2 is called (line 11) a solution using the nominal bounds if formed. The solutions formed by LEGS2 during its succeeding calls (lines 17, 19 and 21) use BOUNDS/2, BOUNDS/4, and BOUNDS/8. The successive solutions are stored.

2.2.5 Subroutine Flow Anomalies

There are anomalies in the flow which are not departures from the logic, but merely instances where the drafting ground rules are impractical. Since they are few in number they can be individually described.

The first anomaly (in Figure 2-3, line 15 through 21, page 2-13) represents a decision with five alternatives. The triangular decision symbol has its left vertex touching the third vertical lines. This means that all alternatives have the same rank as subroutines which are indented to the third line. For example, the subroutine MNELTC and its subordinate decisions and subroutines should be considered to be moved one indentation space to the left.

The second anomaly (Figure 2-5, line 27, page 2-25; line 7, page2-28 ;Figure 2-7, line 31, page 2-37; line 7, page 2-29) is also a triangle calling for a decision, this time with four alternatives, as to which atmosphere model is to be used. The alternatives all have the same rank as the subroutines which are indented as far as the line which touches the left vertex of the decision triangle. Any subroutines called by the alternatives subroutines should be indented one more space to the right.

The third anomaly (Figure 2-5, lines 15 through 27, page 2-26; lines 9 through 21,page 2-38) is an array of boxes which represent decision strings. Regardless of the path taken, all decisions are considered the same rank as the first decision which is represented by the large diamond.

The last anomaly (Figure 2-5, line 28, page 2-27) is a decision with five alternatives. Again, all are of rank equal to the subroutines which are indented as far as the left vertex of the decision triangle. It should be noted that here alternatives 5 and 6 continue on to REWT, alternative 7 returns to SETIC for another trip through, and alternatives 8 and 9 pass through a decision before continuing.

2.3 SUBROUTINE FLOW CHARTS

2.3.1 ESPOD Subroutine Flow Charts

2.3.1.1 ESPOD Guide Flow

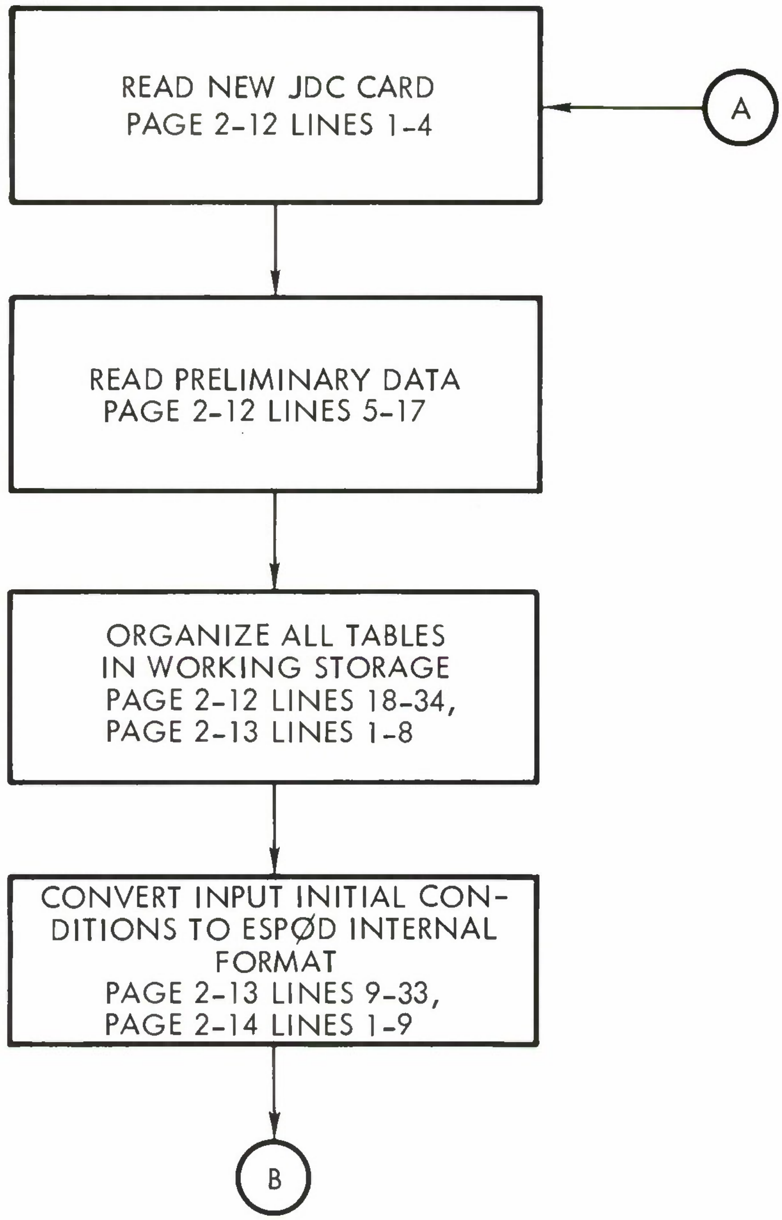


Figure 2-2. ESPØD Guide Flow

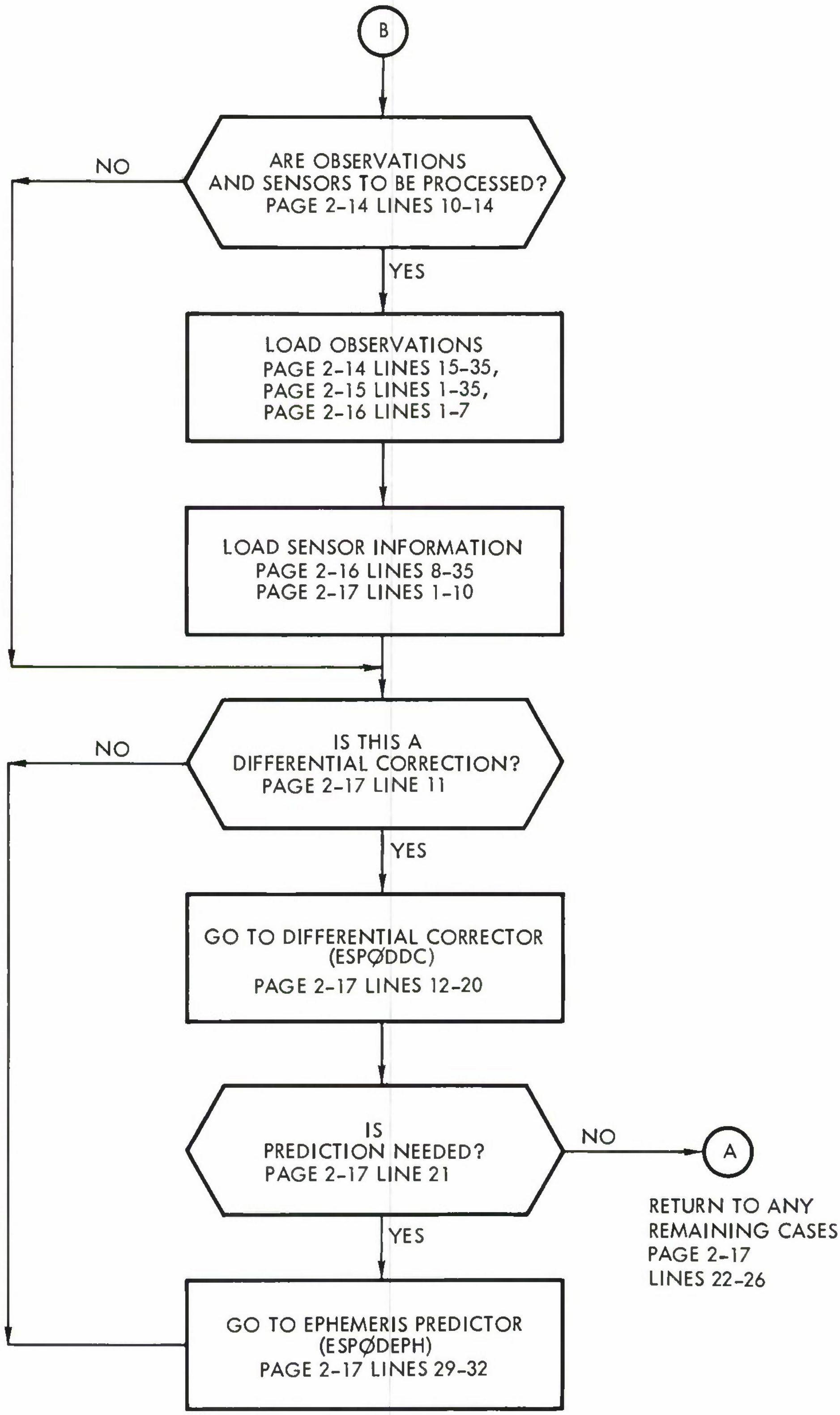


Figure 2-2. ESPØD Guide Flow (Continued)

2.3.1.2 ESPØD Subroutine Flow

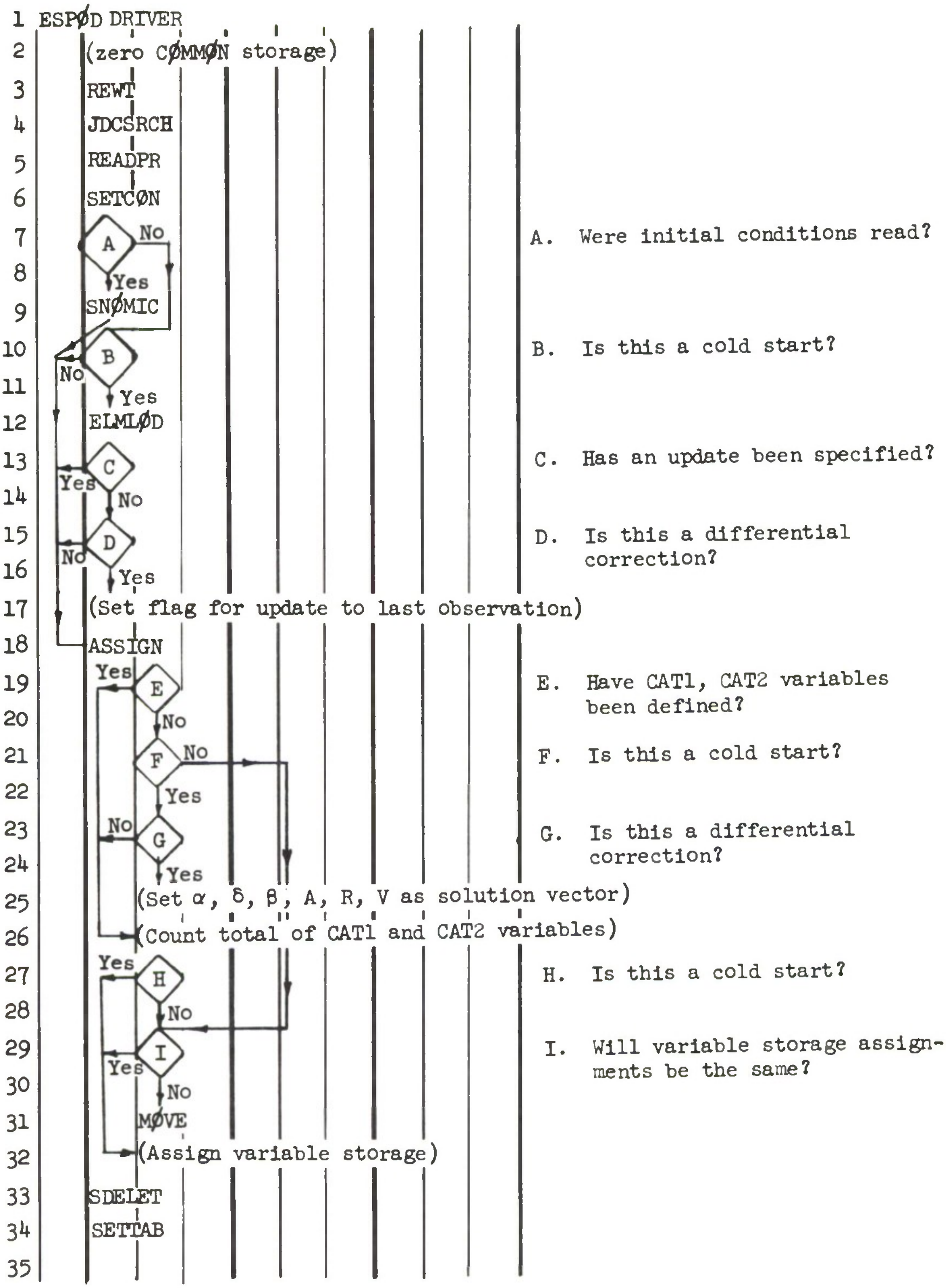
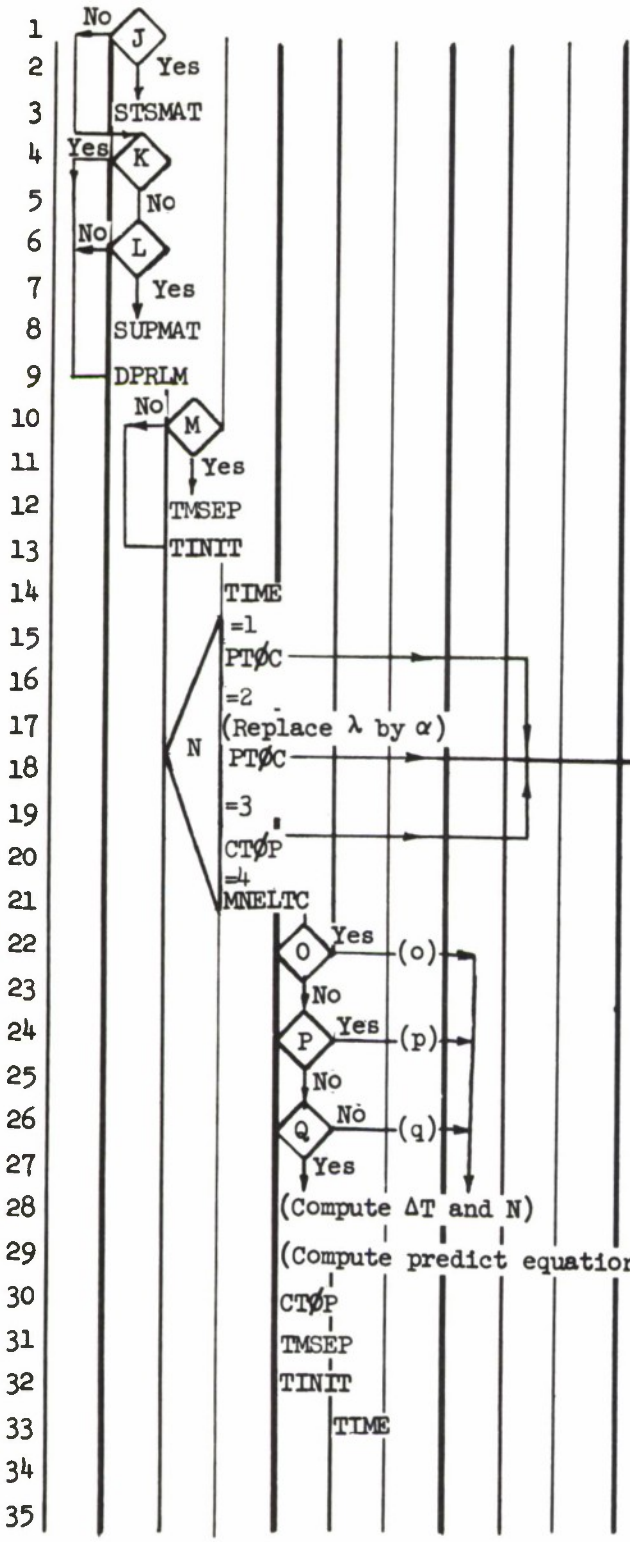


Figure 2-3. ESPØD Subroutine Flow Chart



XX To page 2-14, line 5

J. Was S-matrix read?

K. Is this a differential correction?

L. Is this an update?

M. Are SPADATS 7-card mean elements input?

N. Test initial conditions input flag (D type)

O. Are elements from SEAI?

o. (Compute a_{xn}, a_{yn}, L and N)

P. Osculating elements?

p. (Compute P_o, e_o, a_o, q_o, i_o, n_o Ω_o, P_N, C_N, ω_o)

Q. K-25 mean elements?

q. (Compute a_o)

Figure 2-3. ESPØD Subroutine Flow (Continued)

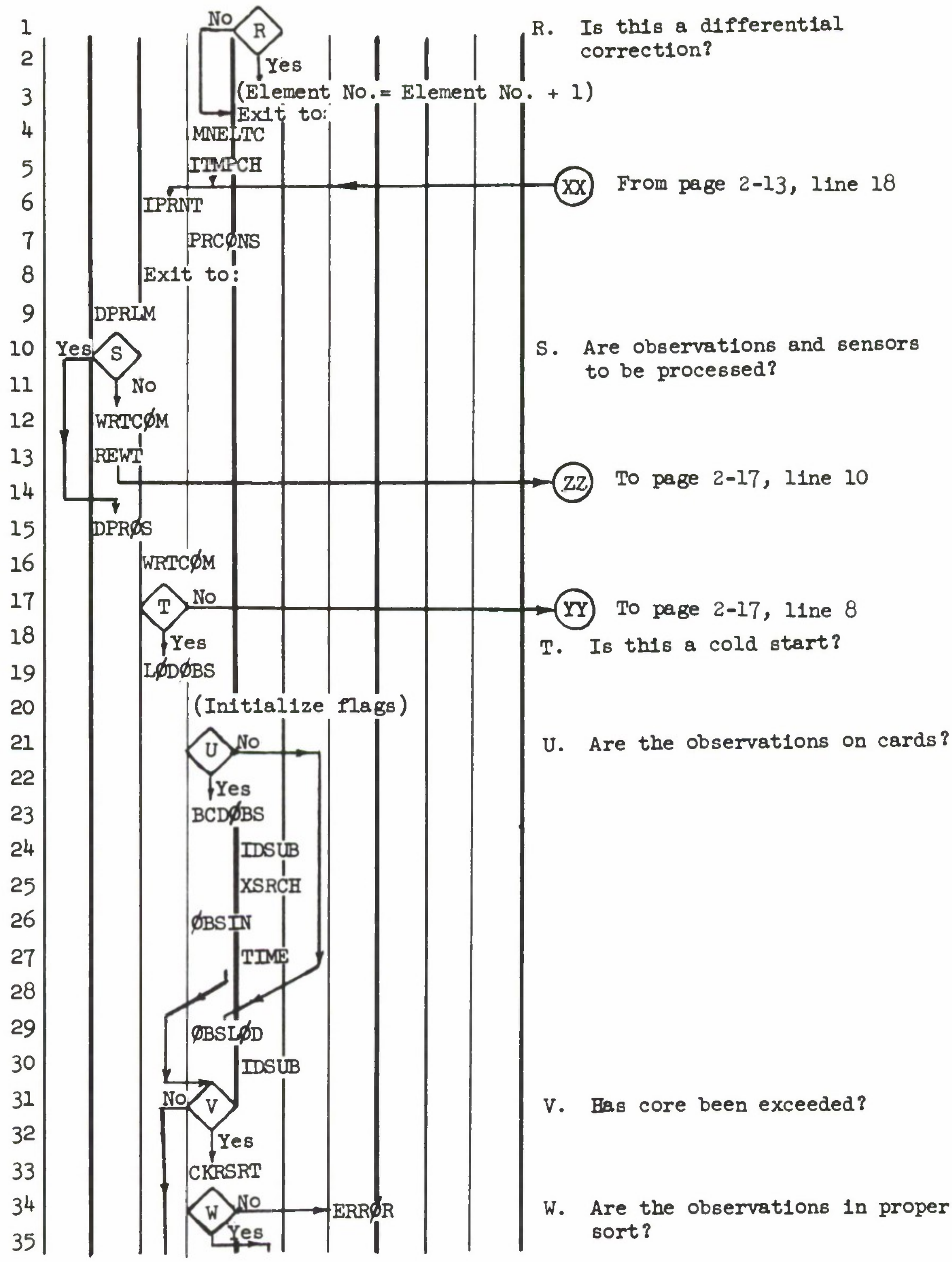
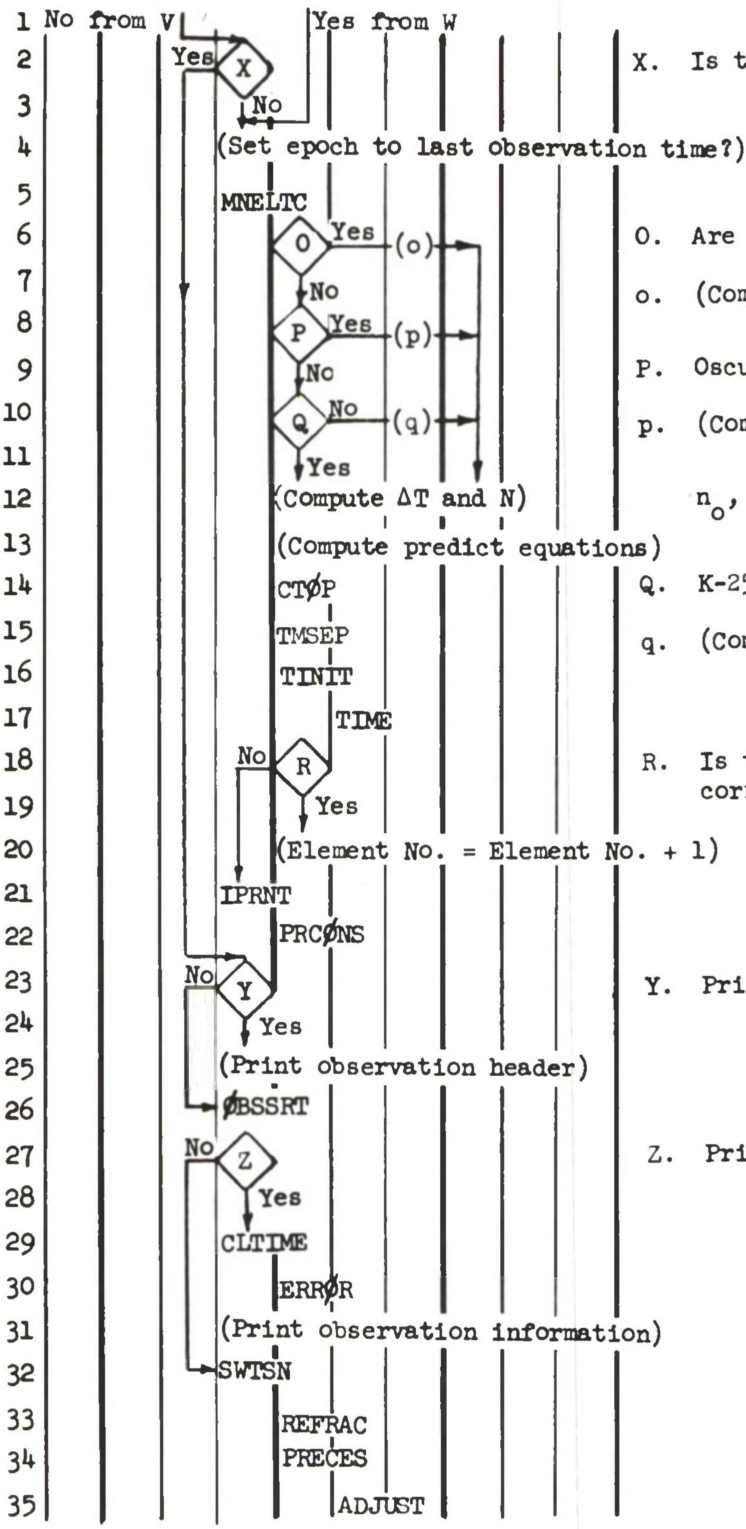


Figure 2-3. ESPØD Subroutine Flow (Continued)



X. Is the epoch set?

O. Are elements from SEAI?

o. (Compute a_{xn}, a_{yn}, L and N)

P. Osculating elements?

p. (Compute P_o, e_o, a_o, q_o, i_o, n_o, Ω_o, P_N, C_N, ω_o)

Q. K-25 mean elements?

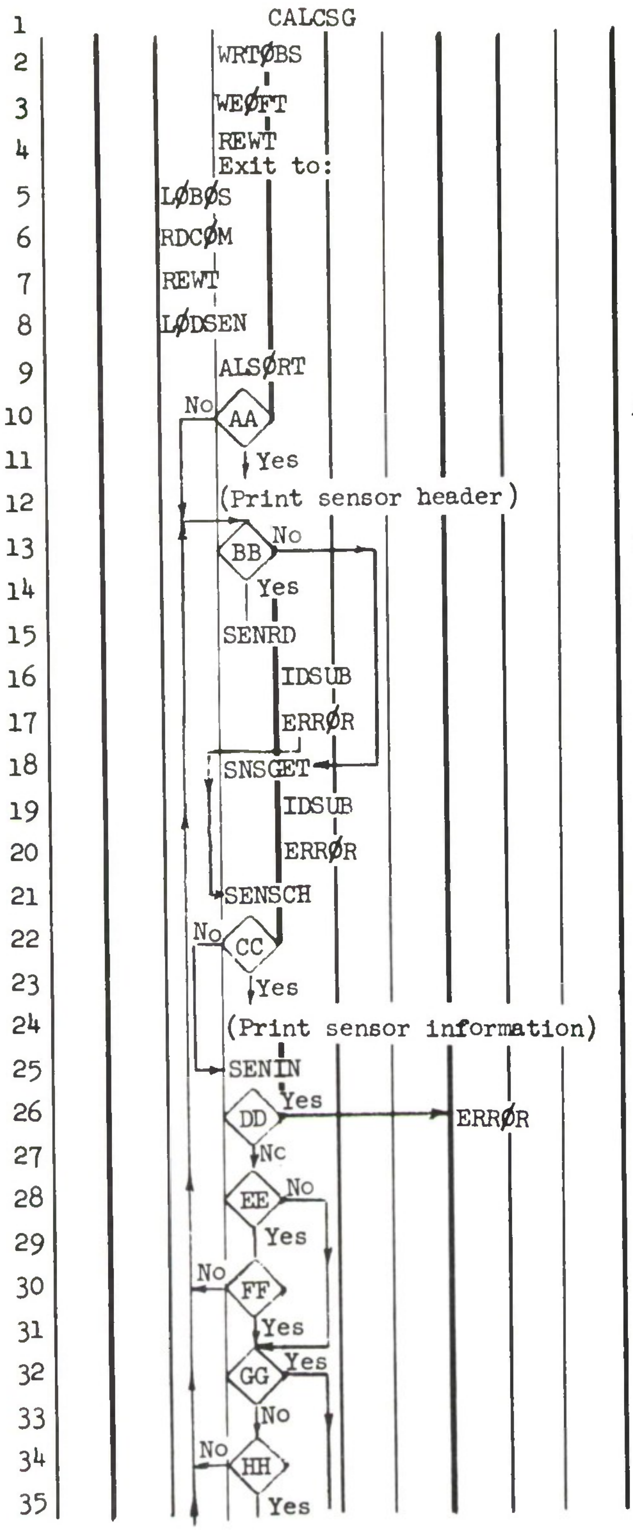
q. (Compute a_o)

R. Is this a differential correction?

Y. Print observation header?

Z. Print observation information?

Figure 2-3. ESPØD Subroutine Flow (Continued)



AA. Is the sensor header to be printed?

BB. Are the sensors on cards?

CC. Are the sensors to be printed?

DD. Do sensors overflow core?

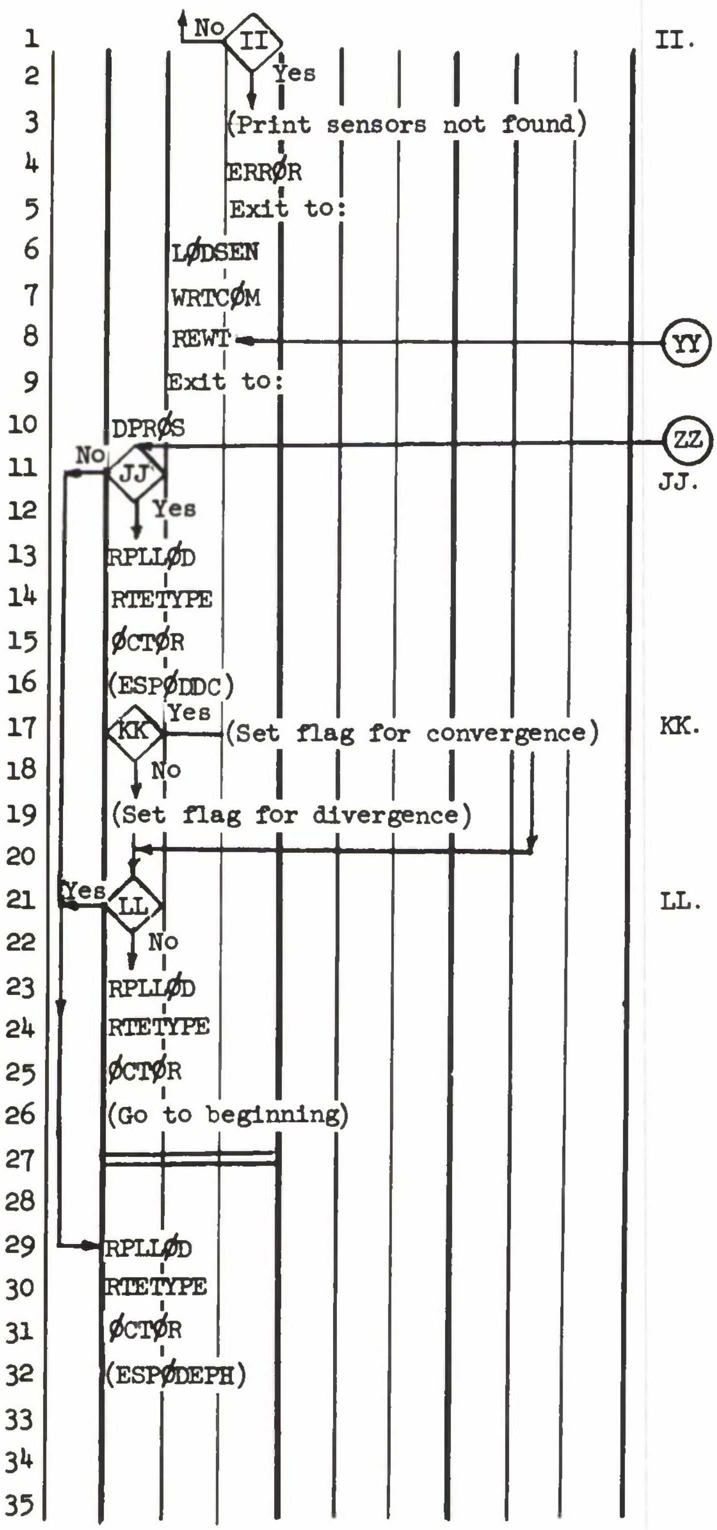
EE. Are sensors on cards?

FF. Are the sensor cards read?

GG. Are required sensors found?

HH. Is the tape mode set? If no, set mode for sensors on tape.

Figure 2-3. ESPØD Subroutine Flow (Continued)



II. Is the sentinel block found?

YY From page 2-14, line 17

ZZ From page 2-14, line 14

JJ. Is this a differential correction?

KK. Convergence?

LL. Is the postprocessor needed?

Figure 2-3. ESPØD Subroutine Flow (Continued)

2.3.1.3 Glossary of ESPØD Subroutines

Subroutine	Functional Description
ADJUST	Updates right ascension, declination measurements to equinox of integration
ALSØRT	Sorts the list of desired sensors alphanumerically
*ASIN	Arcsine routine
ASSIGN	Assigns variable storage
*ATNQF	Arctangent routine
BCDØBS	Reads one observation card from input tape
CALCSG	Calculates and stores sigma table entries
CKRSRT	Checks data in core; is in reverse time sort
CLTIME	Converts an input time into its Gregorian representation
CTØP	Converts Cartesian to polar coordinates
DPRLM	Data initializing (partial)
DPRØS	Driver for loading observation and sensor cards
DRIVER	ESPØD main control
ELMLØD	Control package for loading orbital elements from next tape
ERRØR	General error routine
*EXIT	Empties output buffers and goes to next case
IDSUB	Substitutes in the register for the sensor ID
IPRNT	Prints header page
ITMPCH	Punches the initial epoch time when mean element cards are input

*Designates subroutines used, but not listed in flow because of their routine function

Subroutine	Functional Description
JDCSRCH	Searches for JDC card
*LINES	Ejects paper and print heading at top of page
LØDØBS	Control package for loading observation cards from input tape
LØDSEN	Control package for loading sensor cards from input tape
MØVE	Moves storage in block
MNELTC	Converts SPADATS mean elements to Cartesian
ØBSIN	Moves observations from buffer to permanent storage
ØBSLØD	Loads observations from tape into core
ØBSSRT	Sorts observations to time sequence
ØCTØR	Included in B2 system
*PIMØD	Gets positive argument of an angle in radians between 0 and 2π
PRCØNS	Prints program constants, sensor types, and sensor sigmas
PRECES	Sets up information for ADJUST
PTØC	Converts polar to Cartesian coordinates
RDCØM	Reads common block from observation tape
READPR	Reads preliminary data
REFRAC	Computes tropospheric refraction correction
RPLLØD	Included in B2 system
REWT	Rewinds observation tape
RTETYPE	Included in B2 system

*Designates subroutines used, but not listed in flow because of their routine function

Subroutine	Functional Description
SDELET	Moves deletion list from buffer to permanent storage
SENIN	Moves sensor data from buffer to permanent storage
SENRD	Reads one sensor card from input tape
SENSCH	Searches sensor table
SETCØN	Set constants for program
SETTAB	Set tables concerning solution vector in variable storage
SNØMIC	Moves nominal conditions from buffer to permanent storage
SNSGET	Loads sensor information from tape
STSMAT	Moves input update matrix from buffer to permanent storage
SWTSN	Monitors weight assignments, refraction, precession, in observations
TIME	Converts Y, M, D, H, M, S to Julian date: days plus fraction
TINIT	Sets up initial time, computes a_{go}
TMSEP	Modulates initial times and sets up permanent storage
*UNPAKSN	Unpacks integers in two cells and stores them in four cells
WEOFT	Writes ending sentinel block on observation tape
WRTCØM	Writes CØMMØN block from observation tape
WRTOBS	Generates observation tape
XSRCH	Reads 99 card images

*Designates subroutines used, but not listed in flow because of their routine function

2.3.2 ESPØDDC Subroutine Flows

2.3.2.1 ESPØDDC Guide Flow

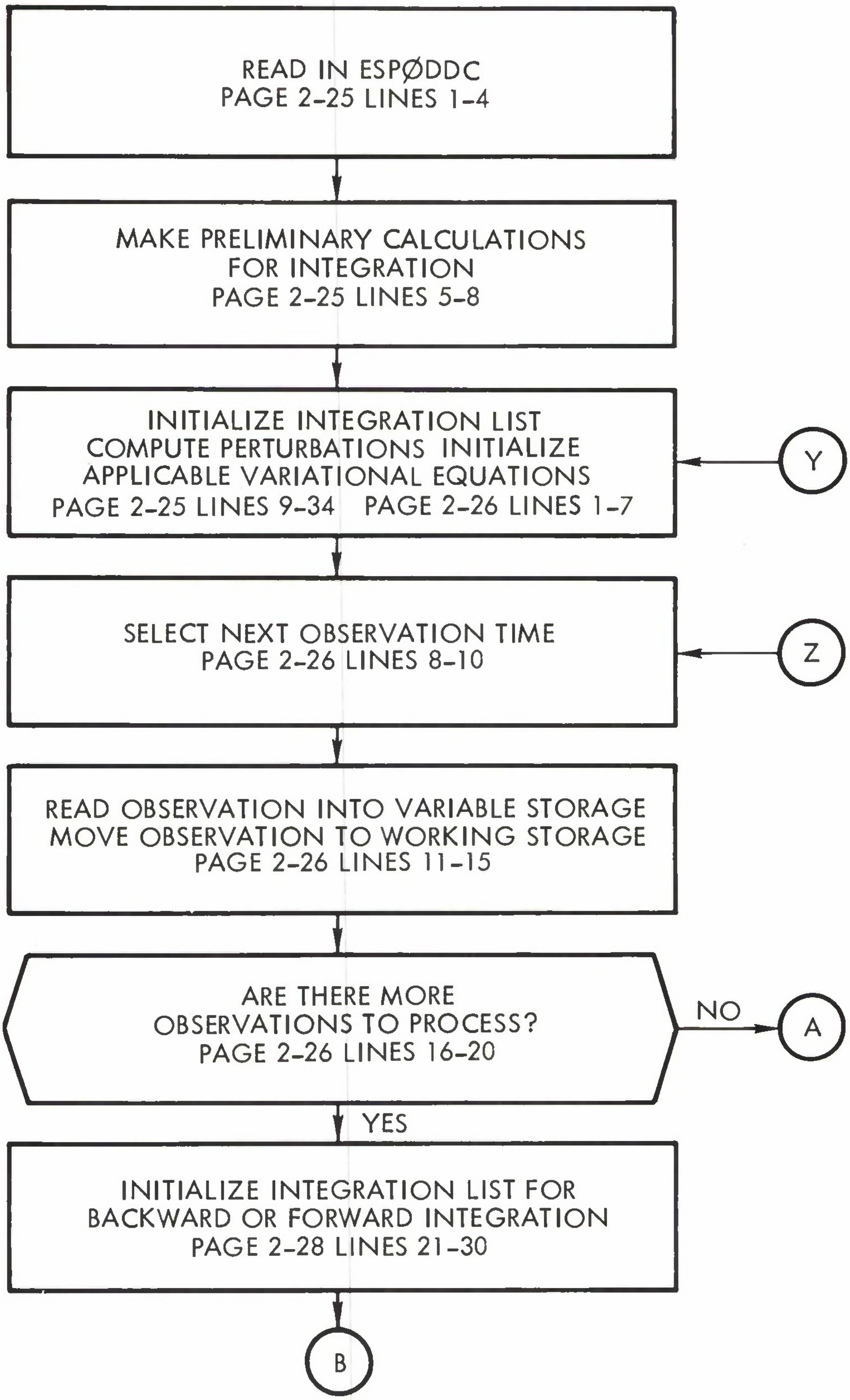


Figure 2-4. ESPØDDC Guide Flow

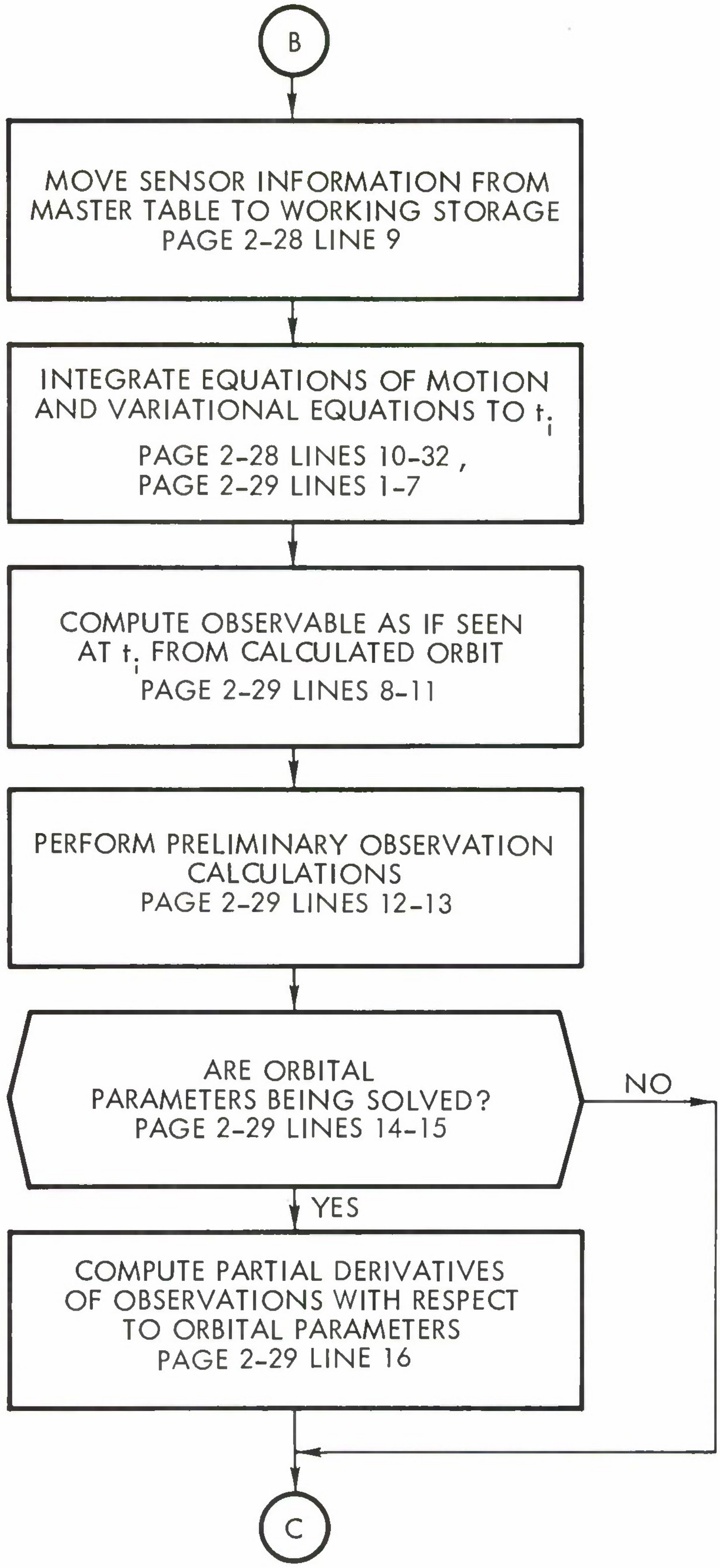


Figure 2-4. ESPØDDC Guide Flow (Continued)

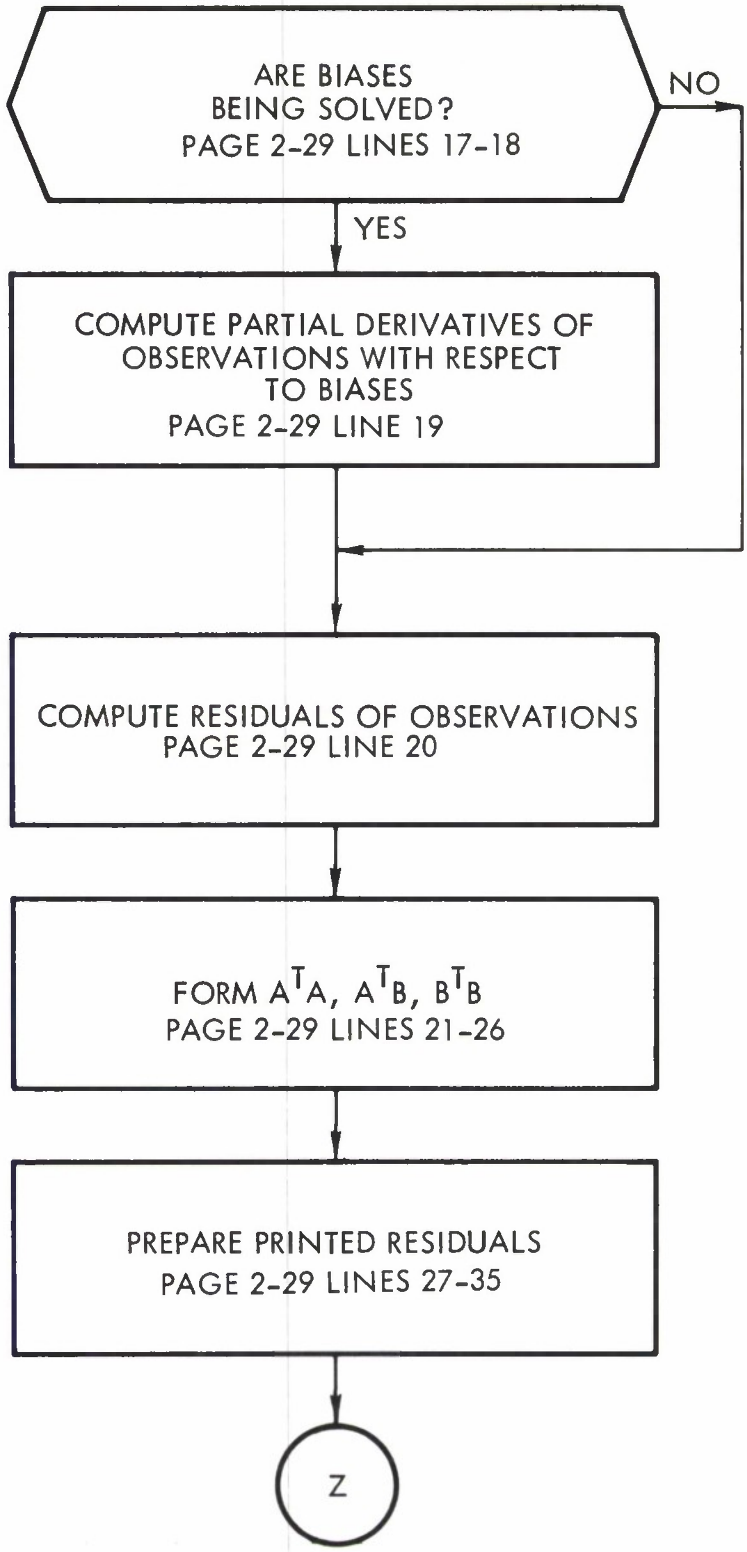


Figure 2-4. ESPØDDC Guide Flow (Continued)

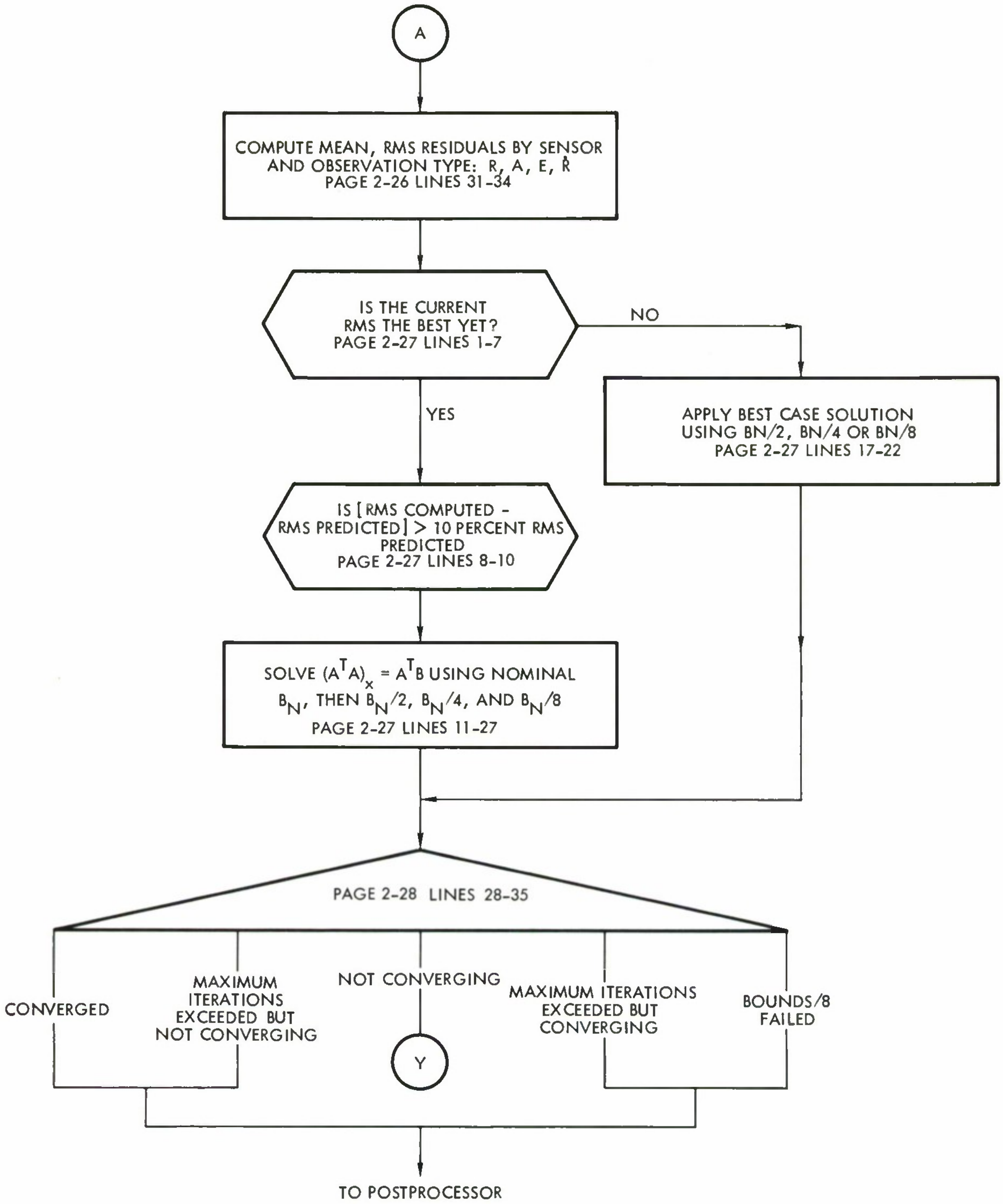


Figure 2-4. ESPØDDC Guide Flow (Continued)

2.3.2.2 ESPØDDC Subroutine Flow

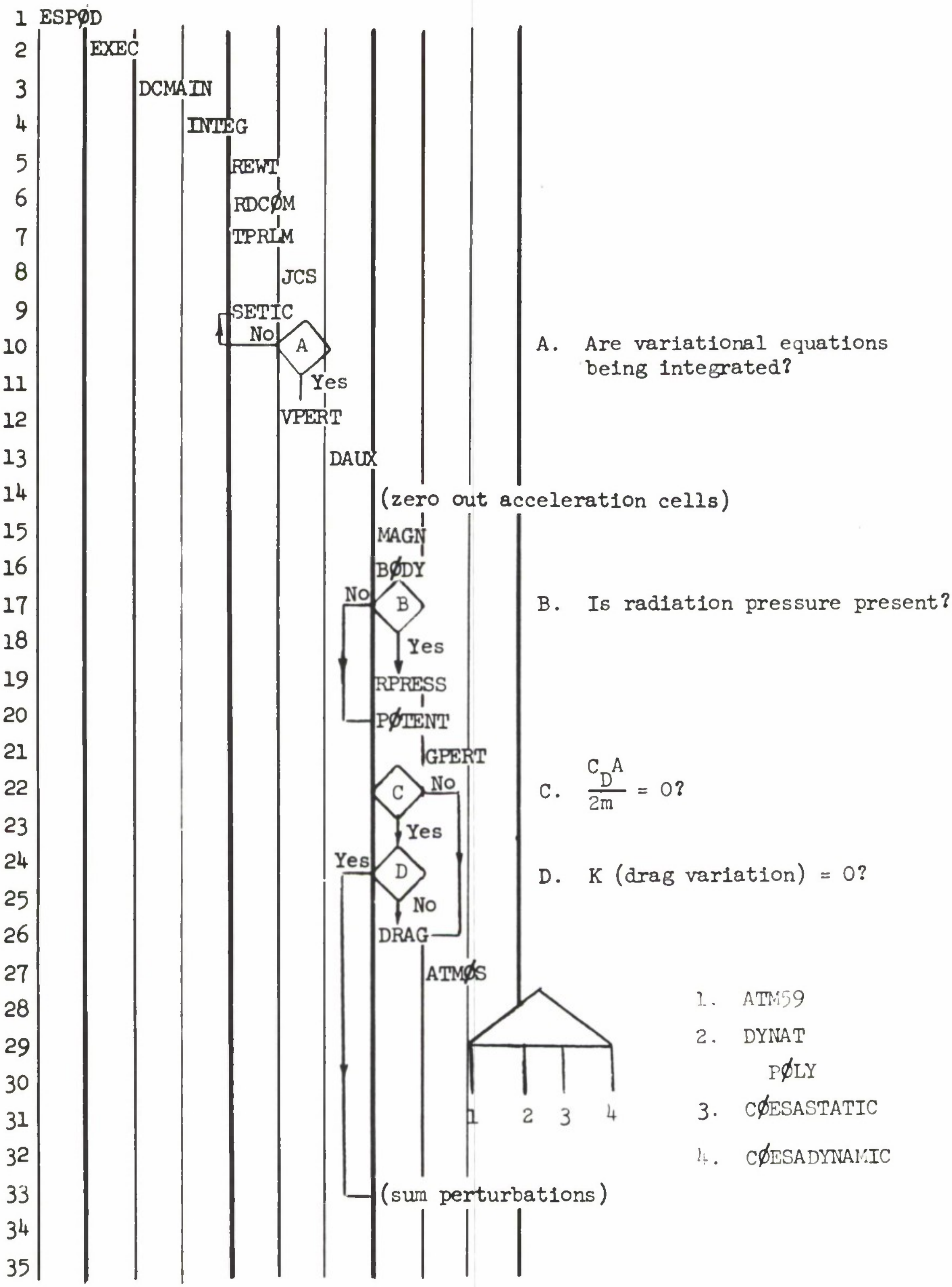


Figure 2-5. ESPØDDC Subroutine Flow Chart

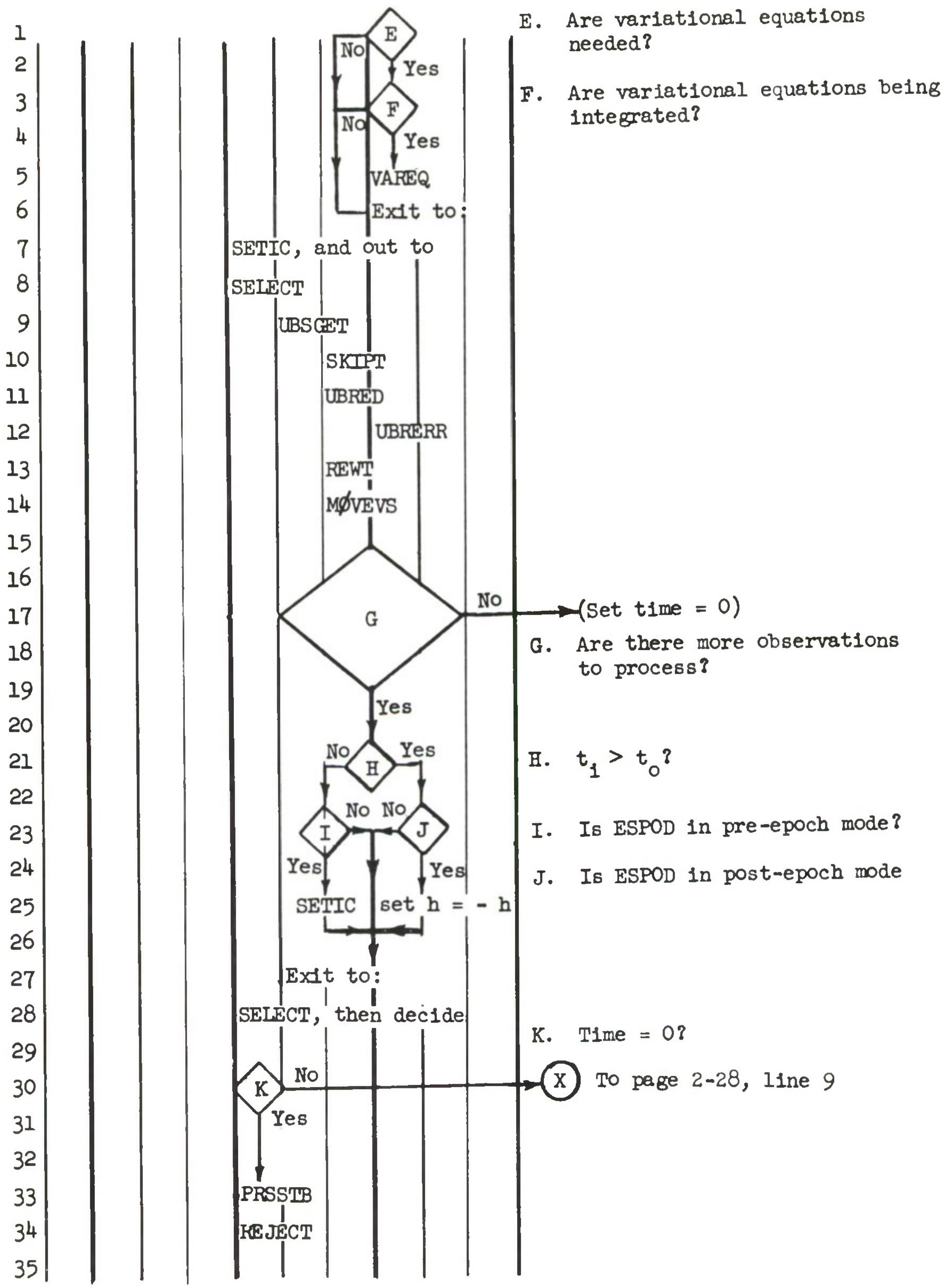
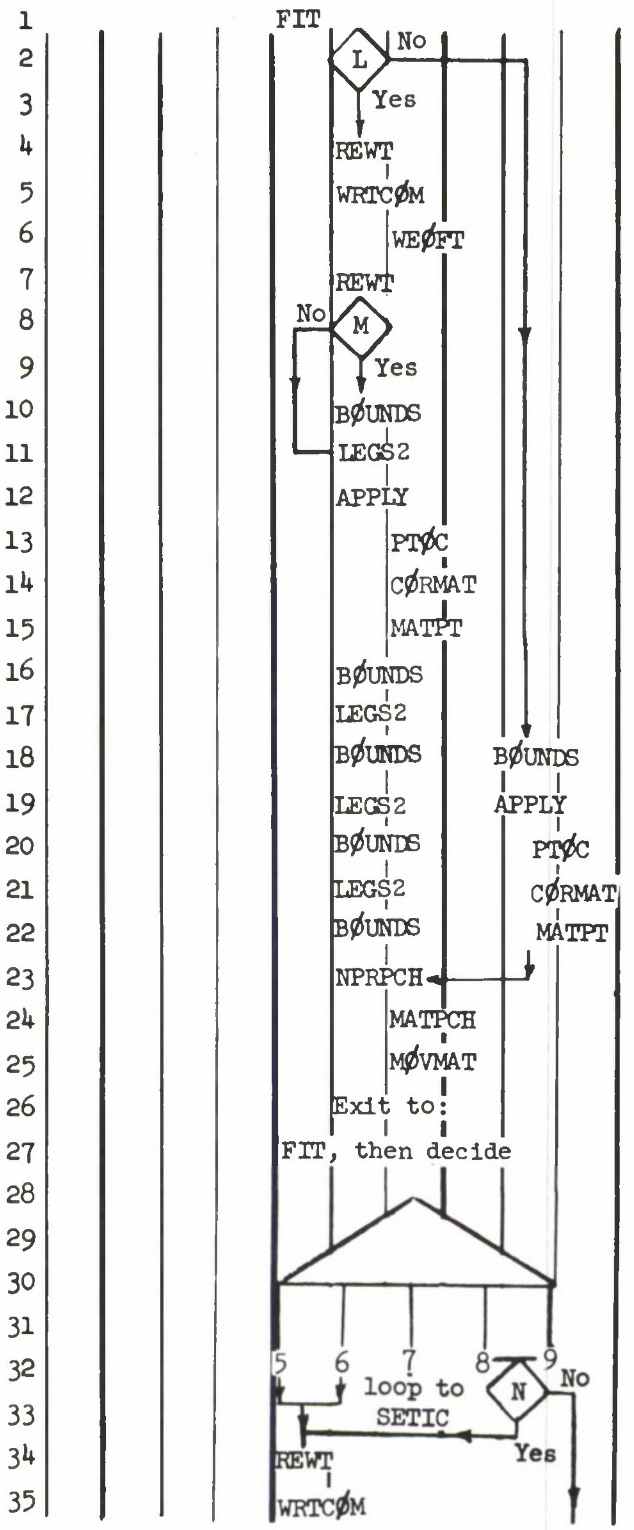


Figure 2-5. ESPØDDC Subroutine Flow (Continued)



L. Is the currest RMS the best?

M. Is | RMS computed - RMS predicted | > 10% RMS predicted?

5. Converged

6. Max iterations exceeded and converging

7. Not converging

8. Bounds/8 failed

9. Max iterations exceeded and not converging

N. Write new elements on tape?

Figure 2-5. ESPØDDC Subroutine Flow (Continued)

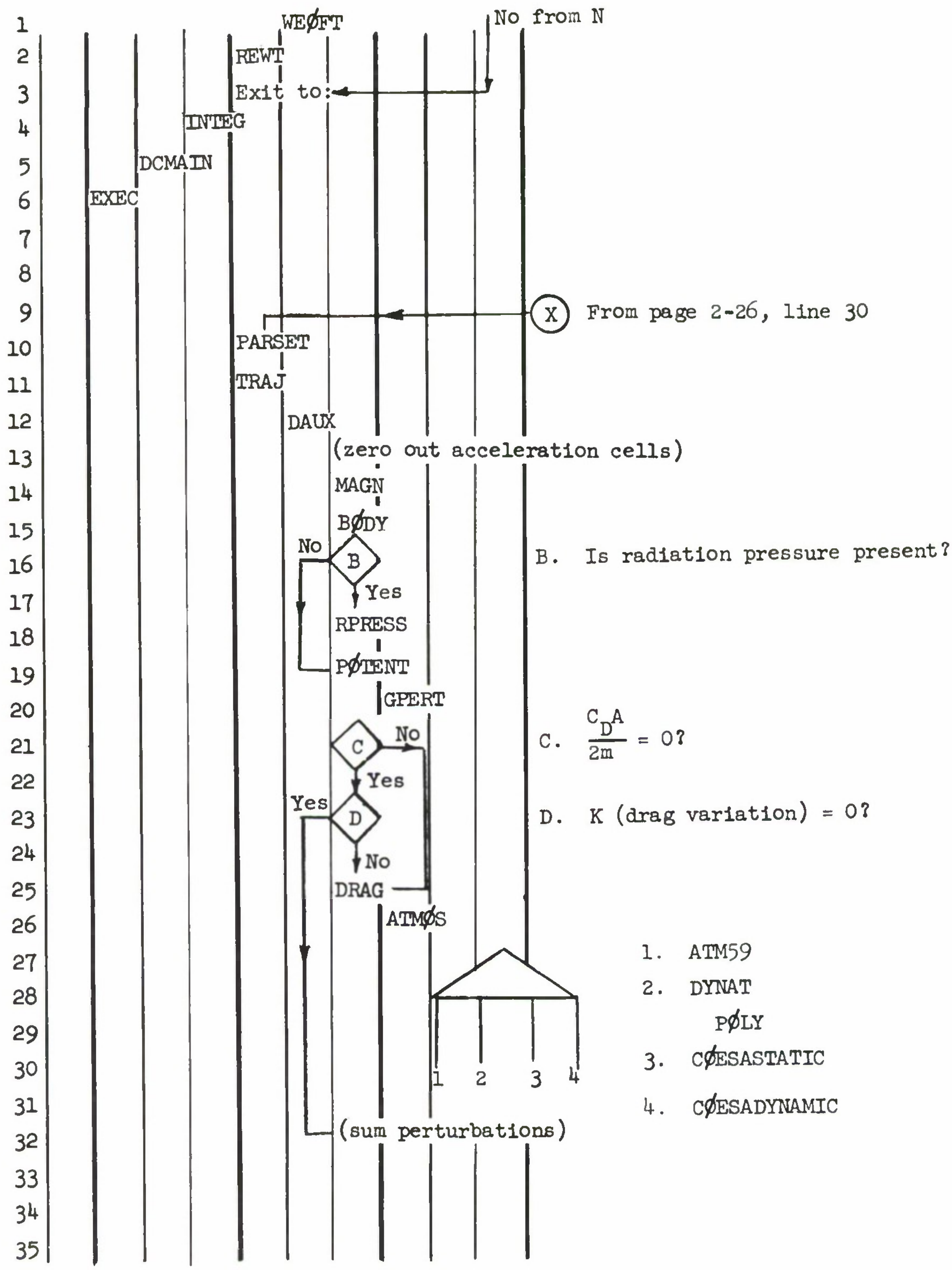


Figure 2-5. ESPØDDC Subroutine Flow (Continued)

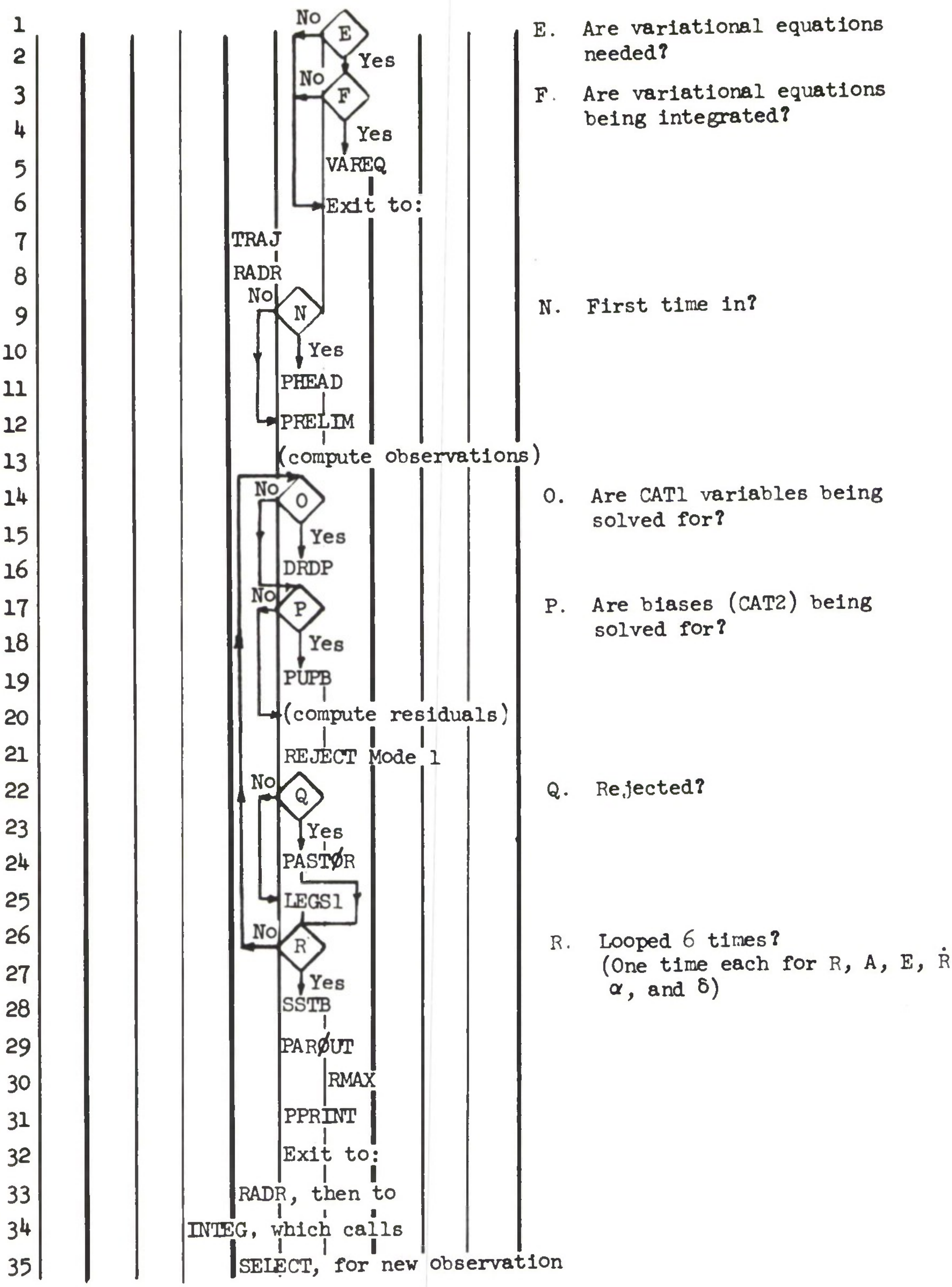


Figure 2-5. ESPØDDC Subroutine Flow (Continued)

2.3.2.3 Glossary of ESPØDDC Subroutines

Subroutine	Functional Description
APPLY	Applies DC solution vector and prints results
*ASIN	Arcsine routine
ATM59	Computes density of a static atmosphere (ARDC 1959 Model)
ATMØS	Driver for density calculation
*ATNQF	Arctangent routine
BØDY	Computes acceleration due to sun and moon
BØUNDS	Scales bounds with given scale factor
CØESA	Computes density for a static or dynamic atmosphere (U. S. Standard 1962)
CØRMAT	Computes correlation (σ and ρ) matrix
DAUX	Driver for evaluating acceleration in integration
*DØN	Computes modifier used in simulated drag variation
*DØT	Computes scalar product
DRAG	Computes drag perturbations
DRDP	Computes partial of observations w. r. t. Category 1 type variables, i. e., α, δ, β, A, r, v, drag
DYNAT	Computes density of a dynamic atmosphere (Paetzold)
FIT	Logic control for DC options
GPERT	Computes acceleration due to Earth's potential
INTEG	Driver for DC package
INTPL	Leads ephemeris tape

*Denotes subroutines used, but not listed in flow because of their routine function

Subroutine	Functional Description
JCS	Sets up two matrices of C's and S's for GPERT
LEGS1	Forms A^TA and A^TB given A and B
LEGS2	Least squares package solves $Ax = B$
*LINES	Ejects page and prints heading at top of page
MAGN	Computes magnitude and $(\text{magnitude})^2$ of 3-D vector
MATPCH	Punches A^TA and $\left(A^TA\right)^{-1}$ at the end of each iteration in a form suitable for input to ESPOD
MATPT	Prints lower triangular matrix
MØVEVS	Moves observation set from variable to working storage
MØVMAT	Moves a triangular matrix from A storage to B storage
*MULT	Multiplies a 3 x 3 matrix by a succession of 1 x 3 vectors
NPRPCH	Punches the ICØND, BISEST, BNDS values at the end of each iteration, in a form suitable for input to ESPØD
*ØUTER	Computes product of column and row vector
PARSET	Initialize station data for partial derivative package
PARØUT	Computes residuals for residuals print
PASTØR	Set up an asterisk or double asterisk for punching to identify a deleted observation
PHEAD	Prints residuals header
*PIMØD	Gets positive argument of an angle in radians between 0 and 2π
PØLY	Evaluates nth order polynomial

*Denotes subroutines used, but not listed in flow because of their routine function

Subroutine	Functional Description
PØTENT	Driver for geopotential model
PPRINT	Prints residuals
PRELIM	Makes preliminary calculations in partial package
PRSSTB	Computes and prints mean, RMS, and number for residuals by sensor and type
PTØC	Converts polar to Cartesian coordinates
PUPB	Computes partial of observation w. r. t. Category 2 variables, i. e., t_b, ϕ_b, λ_b, h_b
RADR	Computes residuals; driver for partials package
RDCØM	Reads CØMMØN block from observation tape
REJECT	Monitors the acceptance or rejection of an observation
REWT	Rewind observation tape
*RMAX	Compare residual quantities with table of maximum values
*RØTRU	Rotates a set of vectors from mean 1950. 0 to true of date
RPRESS	Computes acceleration due to radiation pressure
SELECT	Select next observation time
SETIC	Initialize integration list
SKIPT	Skips CØMMØN block of observation tape after each iteration
SSTB	Accumulates sum, sum of squares, and number of residuals by sensor and data type
TPRLM	Sets up data for integration
TRAJ	Driver for integration program
UBRED	Reads observations into variable storage

*Denotes subroutines used, but not listed in flow because of their routine function

Subroutine	Functional Description
UBSGET	Gets next observation time from variable storage
VAREQ	Computes second derivatives of variational equations
VPERT	Initializes variational equations
WEØFT	Writes an ending sentinel block on observation tape
WRTCØM	Writes CØMMØN block from observation tape
*XCROSS	Performs the cross product of two 3-D vectors
*YHADEC	Compute Y vector when range, hour angle, and declination are given
*YRAE	Compute Y vector when range, azimuth, and elevation are given

*Denotes subroutines used, but not listed in flow because of their routine function.

2.3.3 ESPØDEPH Subroutine Flow Charts

2.3.3.1 ESPØDEPH Guide Flow

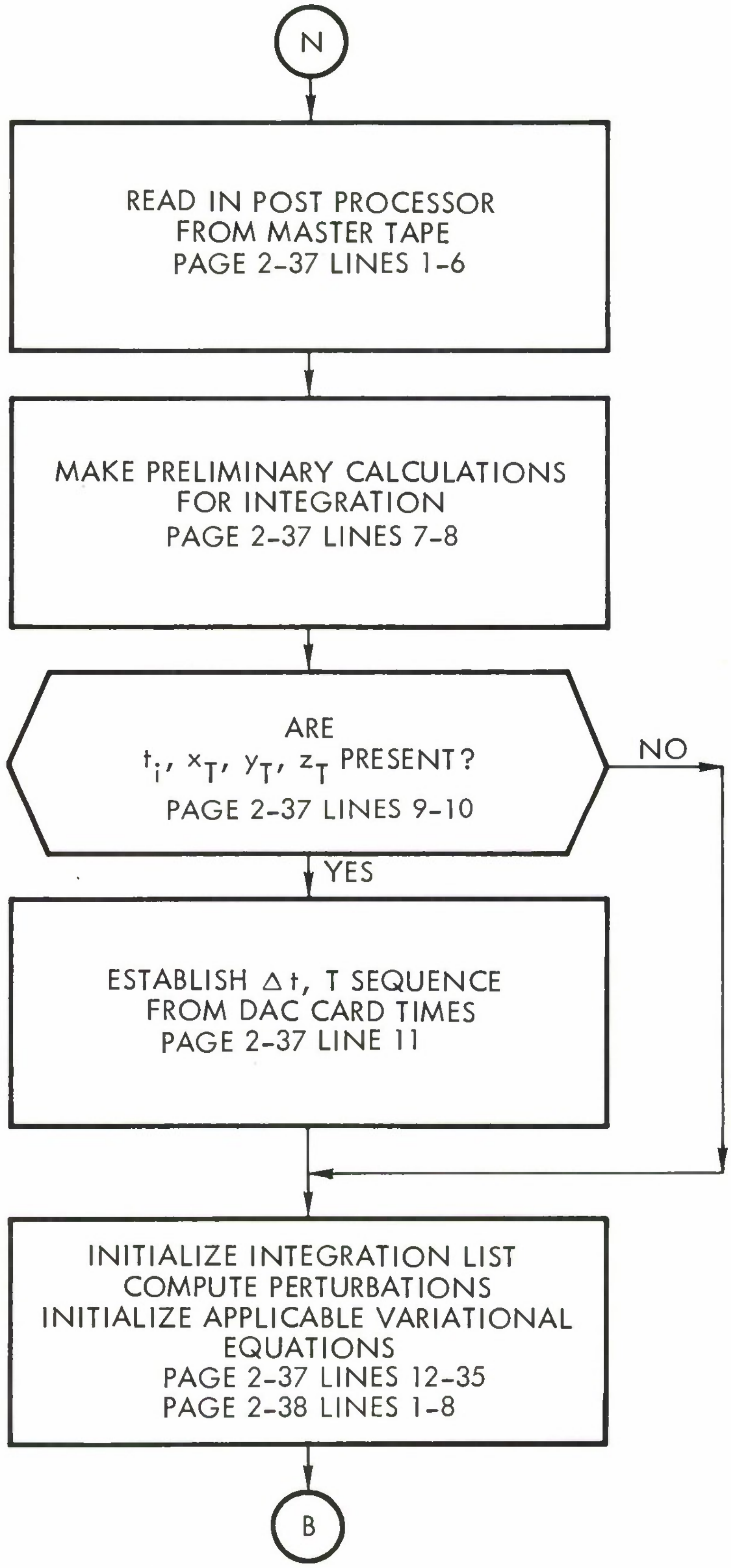


Figure 2-6. ESPØDEPH Guide Flow

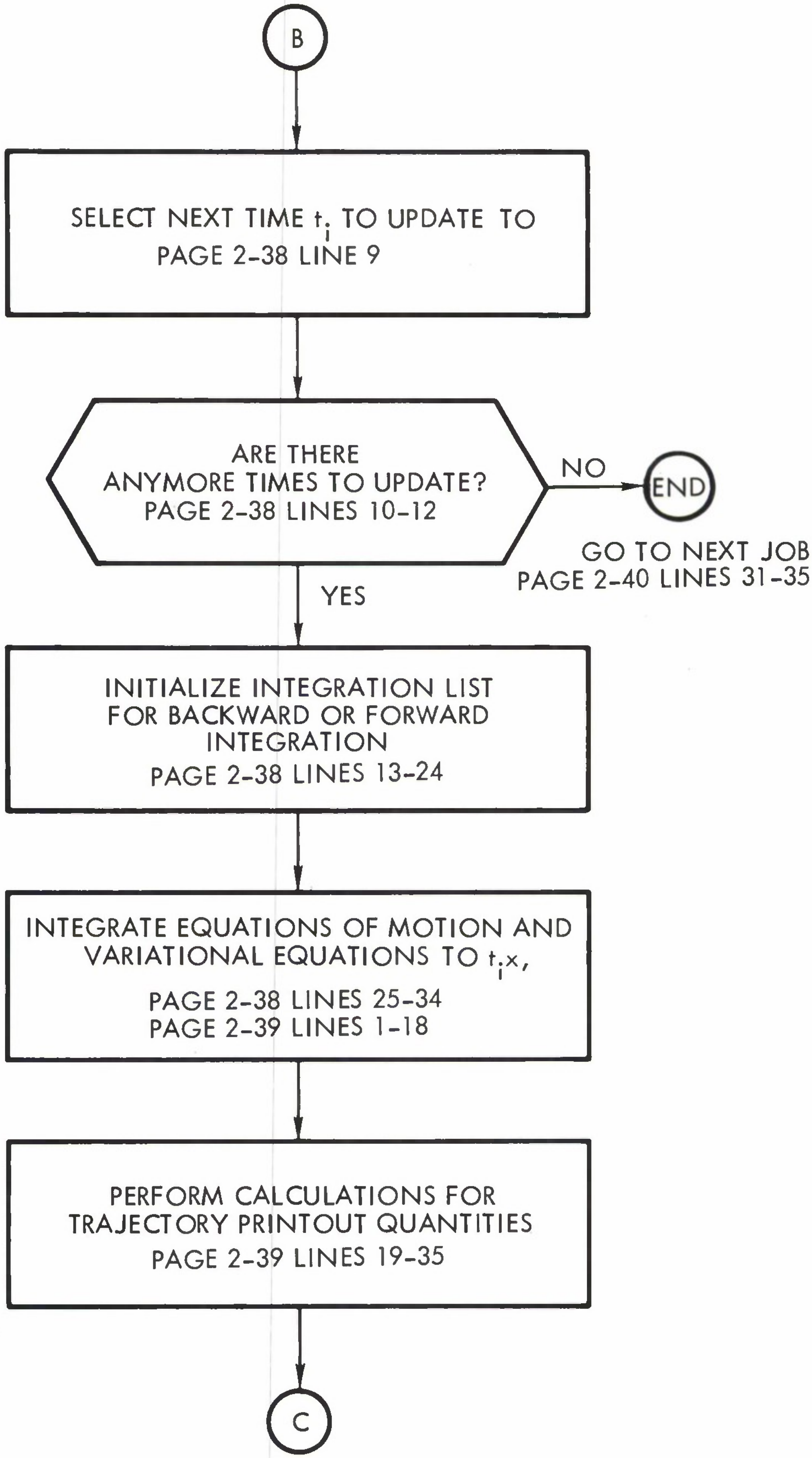


Figure 2-6. ESPØDEPH Guide Flow (Continued)

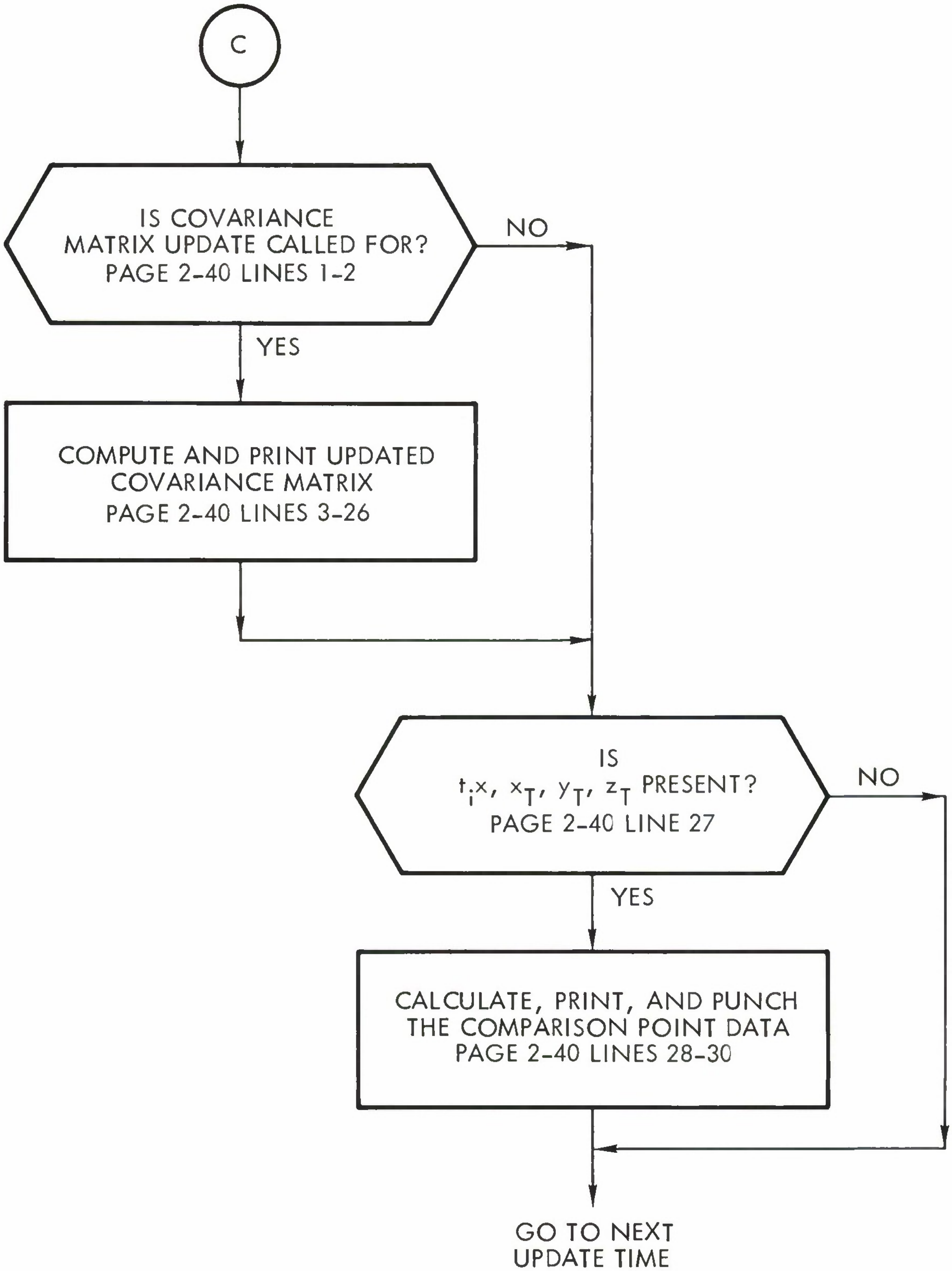


Figure 2-6. ESPØDEPH Guide Flow (Continued)

2.3.3.2 ESPØDEPH Subroutine Flow

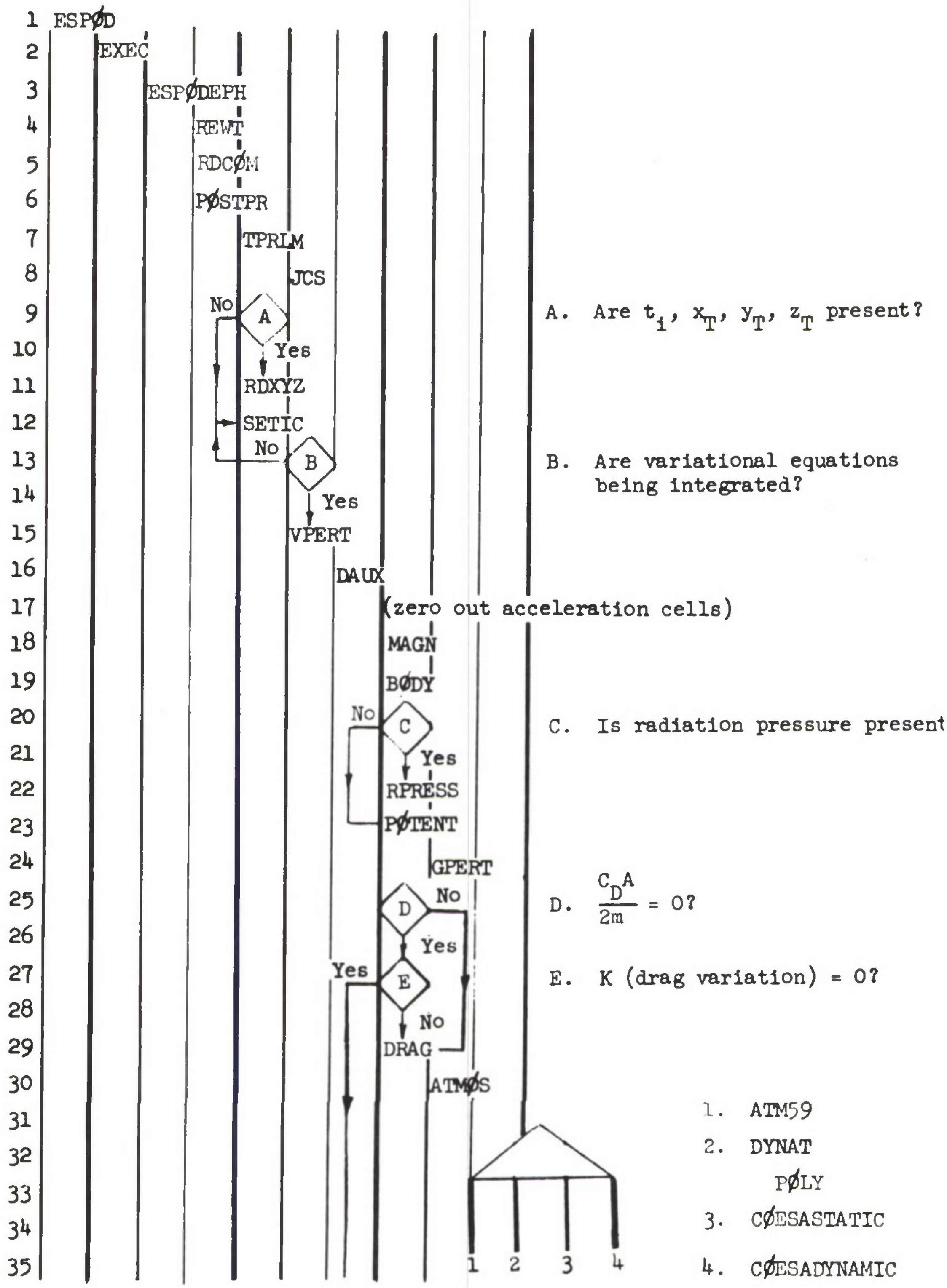


Figure 2-7. ESPØDEPH Subroutine Flow Chart

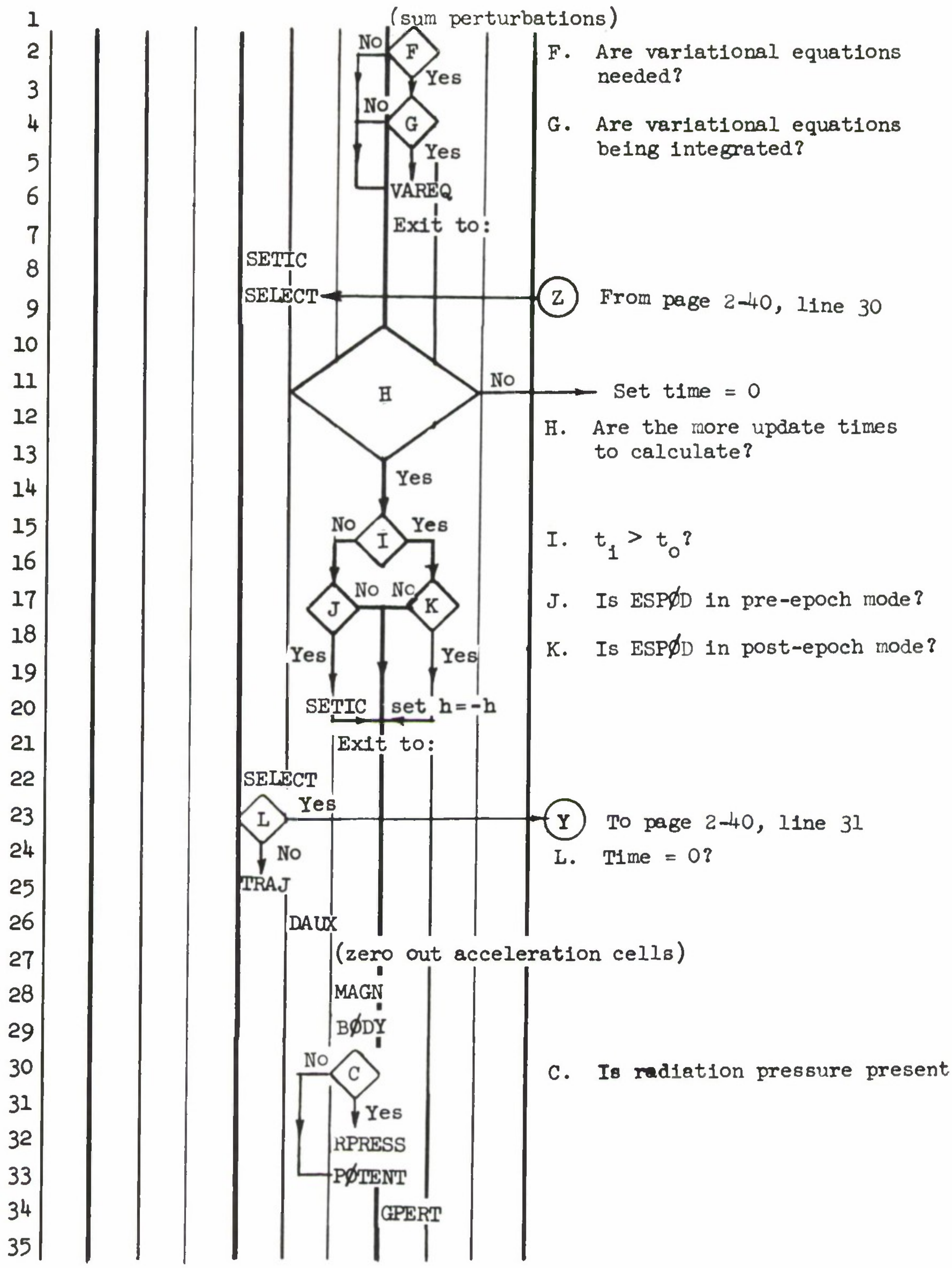


Figure 2-7. ESPØDEPH Subroutine Flow (Continued)

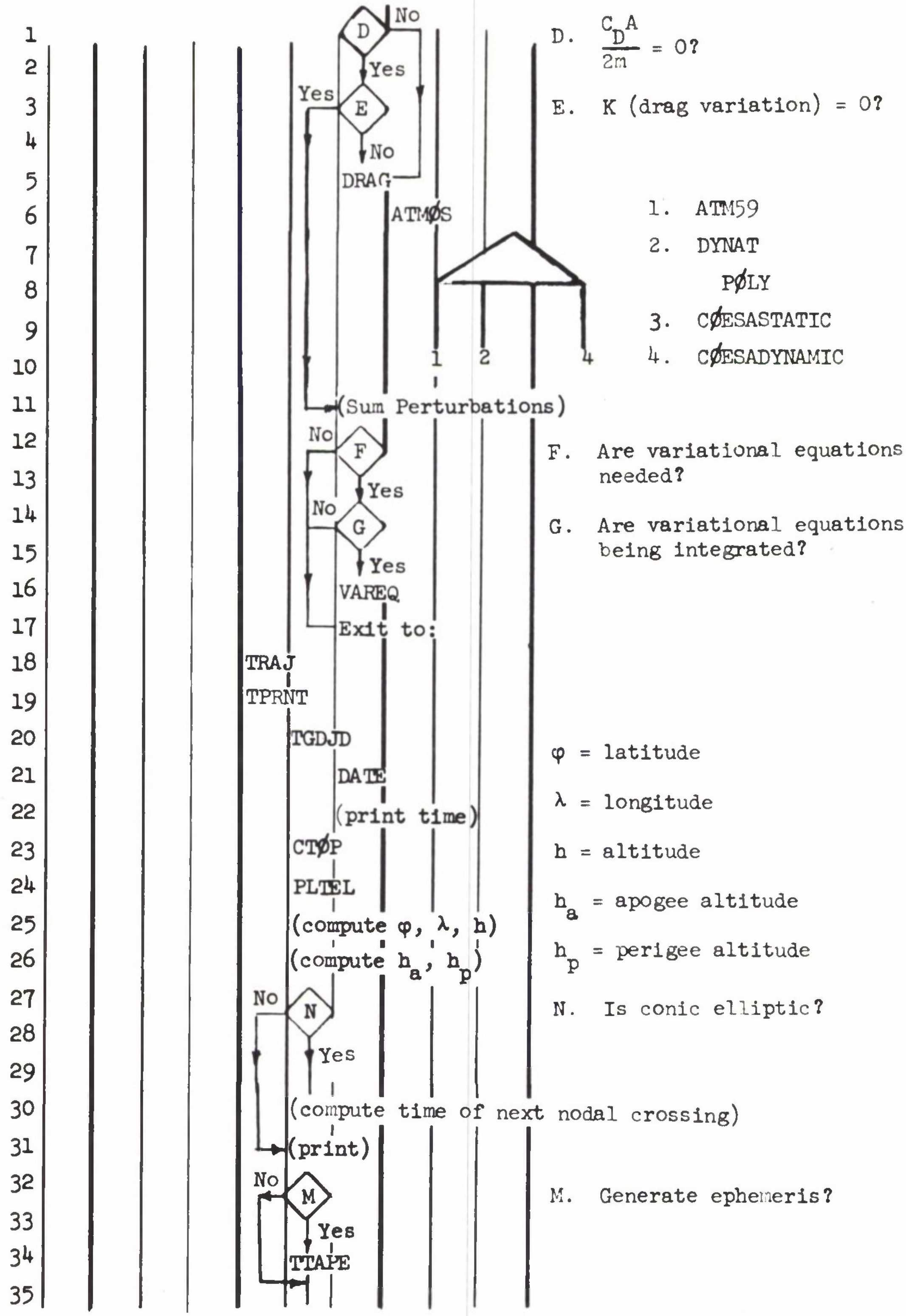


Figure 2-7. ESPØDEPH Subroutine Flow (Continued)

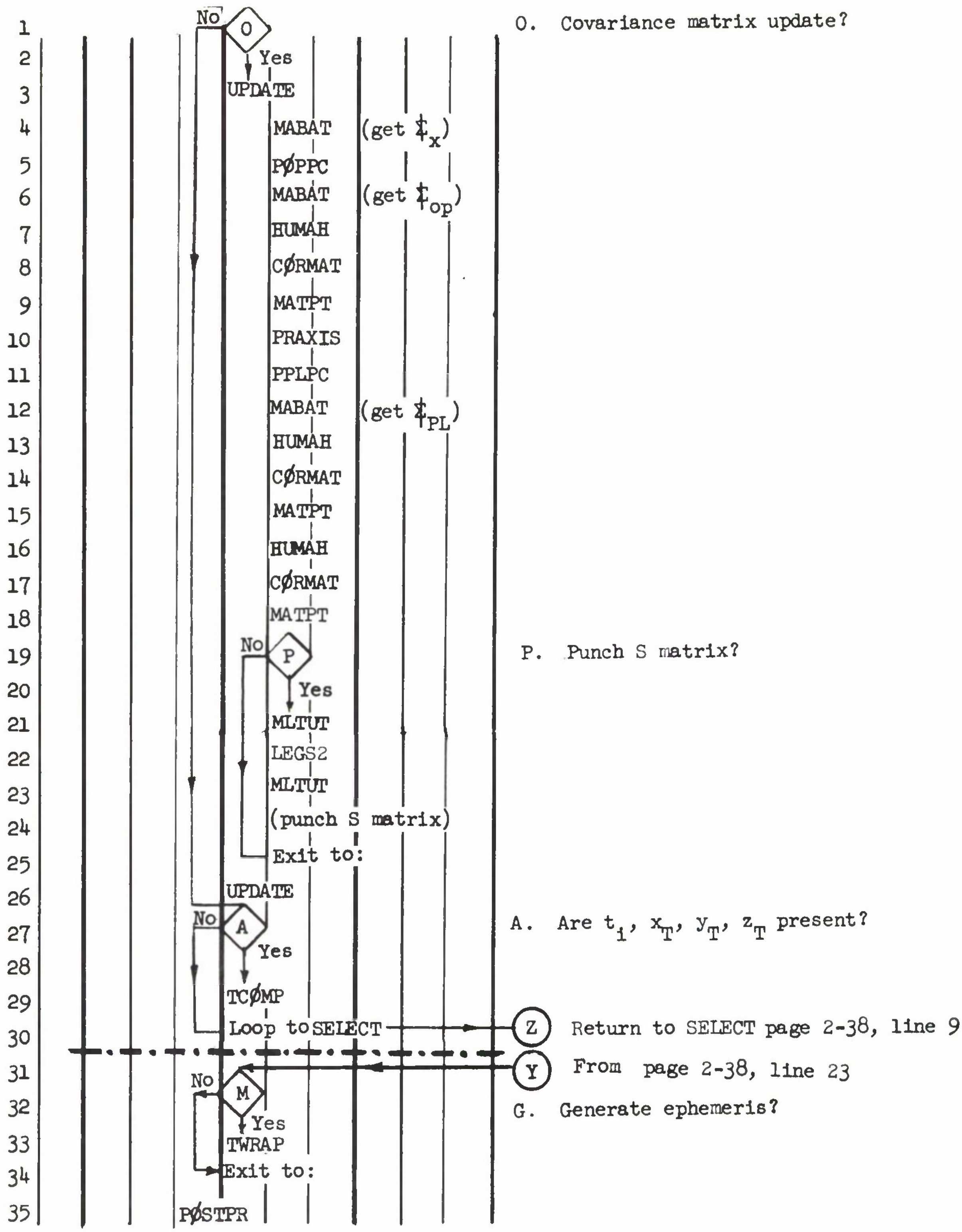


Figure 2-7. ESPØDEPH Subroutine Flow (Continued)

2.3.3.3 Glossary of ESPØDEPH Subroutines

Subroutine	Functional Description
*ASIN	Arscine routine
ATM59	Computes density of static atmosphere (ARDC 1959 Model)
ATMØS	Driver for density calculation
*ATNQF	Arctangent routine
BØDY	Computes acceleration due to sun and moon
CØRMAT	Computes correlation (σ and ρ) matrix
CØESA	Computes density of a static or dynamic atmosphere (U.S. Standard 1962)
CTØP	Converts Cartesian to polar
DATE	Converts time in minutes from 0 hours day of epoch to calendar time
DAUX	Driver for evaluating acceleration in integration
*DØN	Calculates modifier used in simulated drag variation
*DØT	Computes scalar product
DRAG	Computes drag perturbations
DYNAT	Computes density of a dynamic atmosphere (Paetzold)
GPERT	Computes acceleration due to earth potential
HUMAH	Converts vector or matrix from machine units to human units or vice versa
*INTPL	Reads ephemeris tape
JCS	Sets up two matrices C's and S's for GPERT
LEGS2	Least squares package, solves $A^T Ax = A^T b$

*Denotes subroutines used, but not included in flow because of their routine function

Subroutine	Functional Description
MABAT	Multiplies ABA^T where B is lower triangular matrix
MAGN	Computes magnitude and $(\text{magnitude})^2$ of 3-D vector
MATPT	Prints an N x N lower triangular matrix
MLTUT	Converts lower triangular matrix to upper triangular matrix
*MULT	Multiplies a 3 x 3 matrix by a succession of 1 x 3 vectors
*ØUTER	Computes product of a column and row vector
*ØUTPT	Punches sets of x_T, y_T, z_T, t_D, t_{Df}
*PIMØD	Gets positive argument of an angle in radians between 0 and 2π
PLTEL	Converts polar to elements (indeterminacy free and classical)
PØLY	Evaluates nth order polynomial
PØPPC	Sets up rotation from Cartesian to orbit plane coordinates
PØSTPR	Driver for postprocessor (ESPØDEPH)
PØTENT	Driver for geopotential model
PPLPC	Computes partial of ADBARV w. r. t. Cartesian
PRAXIS	Calculates yaw, pitch and roll rotations of the principal axes of the error ellipsoid from U, V, W coordinate axis
RDXYZ	Reads x_T, y_T, z_T, t_I from TTY generated cards
REWT	Rewinds observation tape
*RØTRU	Rotates a set of vectors from mean of 1950.0 to true of date

*Denotes subroutines used, but not included in flow because of their routine function

Subroutine	Functional Description
RPRESS	Computes accelerations due to radiation pressure
SELECT	Select next time to update to
SETIC	Initialize integration list
TCØMP	Compare $\left\|(x) - x, (y) - y, (z) - z\right\|$ with ϵ
TGDJD	Computes Julian date to calendar date from integration time and prints
TPRLM	Sets up data for integration
TPRNT	Prints trajectory print
TRAJ	Integrates the equations of motions and variational equations of motions and variational equations to a specified time
TTAPE	Generates ephemeris tape ($x, y, z, \dot{x}, \dot{y}, \dot{z}$, vs T)
TWRAP	Wraps up ephemeris tape generated by TTAPE
UPDATE	Driver for covariance matrix update logic
VAREQ	Computes second derivatives of variational equation
VPERT	Initializes variational equations
*XCRØSS	Performs the cross product of two 3-D vectors

*Denotes subroutines used, but not included in flow because of their routine function

3. ESPOD CONVENTIONS

This section describes the arrangements of arrays and variables in core storage and the magnetic tape formats. In addition, the conventions used for handling the solution vector and the ESPOD units are included.

3.1 VARIABLE STORAGE

Arrays (vectors, matrices, etc.) associated with a differential correction program will vary in length depending on the size of the system of equations being solved. This creates a storage allocation problem for the programmer. One method of solving this problem is to set aside blocks of program storage of fixed dimension, each large enough to handle the maximum case. This method tends to be inefficient in the use of program storage since only a small percentage of each fixed block array will be used on the majority of runs. A second method is to assign, at execution time, each of these arrays a starting position in core and to give the array only the number of cells it needs on this run. This is the approach that has been adopted within ESPOD.

A block of storage of dimension 4700 has been saved at the end of CØMMON for the purpose of storing arrays such as the solution vector, the current estimates of parameters being solved, the normal matrix, the normal matrix inverse, the bounds vector, the scaling vector, the master sensor table, and the observations to be processed by the differential correction program. The observation block is the last block in variable storage. The advantage is that a minimum of 128 words is all that is required (i.e., enough core to read one block from the 7 TAPE). However, if more core is available the program will load as many observation blocks from the 7 TAPE as is permissible on a given read cycle. This read cycle will continue until all observations on the 7 TAPE have been processed.

3.1.1 Definition

This 4700 cell block in ESPOD storage has been labeled by two names: VSTR (floating point variable storage) and IVSTR (fixed point variable storage). The two names have been given so that the storage block can contain both fixed and floating point arrays. Starting locations for particular arrays in variable storage are defined by a set of indices N1, N2, N3—which

are computed at run time by subroutine ASSIGN. For example, VSTR (N3) defines the first element of a floating point array, while IVSTR (N2) defines the starting location of a fixed point array. In Section 4, subroutine description, arrays in variable storage will be defined either by their corresponding index (XX) or by the name VSTR (XX) or IVSTR (XX).

3.1.2 Indices of Stored Arrays

The acceptable indices currently in ESPOD and the arrays they define are given below.

To aid in the definition of these arrays, let

n = Total number of parameters being solved for

m = Total number of Category 1 variables (α, δ, β, A, R, v, $C_D A/2m$, K)

s = Number of sensors whose data is involved in the differential correction process

3.1.2.1 List of Indices

	Name	Mode of Array	Array Definition and Dimension
1.	NIDP	Fixed	Defines parameters of Category 1 to be solved for (m cells)
2.	NPRCD	Fixed	Defines parameters of Category 2 to be solved for (n - m cells)
3.	NPBIS	Floating	Defines the current estimates of the Category 2 variables to be solved for. This array has a 1:1 correspondence with elements in the NPRCD array (n - m cells)
4.	NAROW	Floating	Defines the array where one row of the augmented matrix (A, B) is stored (n + 1 cells)
5.	NBDNS	Floating	Defines the set of bounds to be used in the differential correction (n cells)
6.	NPAR	Floating	Defines the current estimates of the total (Category 1 plus Category 2) set of variables being solved for. This array is 2 n cells long. The first n cells contain the set in internal units and the second n cells contain the set in external units. (2*n cells)

	Name	Mode of Array	Array Definition and Dimension
7.	NDPAR1 NDPAR2 NDPAR3 NDPAR4	Floating	Define four arrays where, on each converging pass, the four solution vectors are maintained (4*n cells)
8.	NSCALE	Floating	Vector of scaling factors which are used to convert the solution vector, the current estimate vector, the normal matrix and the inverse of the normal matrix from either internal units to external units or external units to internal units (n cells)
9.	NATA	Floating	Defines array containing the augmented matrix (A^TA, A^TB) as an <u>upper</u> triangular matrix stored by rows and the scaler B^TB in the last cell of the array. $\left[\frac{(n+1)*(n+2)}{2} \text{ cells}\right]$
10.	NR	Floating	Defines array containing the inverse normal matrix $(A^TA)^{-1}$ stored row wise as a lower triangular matrix. The sub-routine LEGS2 also uses portions of this array for temporary storage $\left[\frac{(n+2)*(n+3)}{2} -1\right]$
11.	NIDLED	Fixed	Defines array containing observation deletion table (i.e., inputs of the DELET card). The dimension is dependent on number of entries given by DELET cards (2* count of lists deleted).
12.	NSTAT	Floating	Master sensor table (15*S)
13.	NUBS	Floating	Observation table (no dimension, occupies remainder of available table)
14.	NRTMP	Floating	Used for intermediate handling of (A^TA) and $(A^TA)^{-1}$ matrices. $\left[\frac{n*(n+1)}{2} \text{ cells}\right]$
15.	NSSTB	Floating	Array for summing residuals, residuals squared, and total residual number by sensor and type. (13*S cells)
16.	NSMAT	Floating	Array containing <u>a priori</u> S matrix as an upper triangular matrix stored by rows $\left[\frac{n*(n+1)}{2} \text{ cells}\right]$

3.2 INTERNAL HANDLING OF SOLUTION VECTOR

3.2.1 Category 1 Variables

Category 1 variables are composed of the following eight parameters: α, δ, β, A, R, v, $C_D A/2m$, and K (drag variation). ESPOD is capable of solving for this set of parameters or for any subset of these parameters. The CAT1 input card provides the analyst the opportunity to select a set of these parameters for solution. The program must then be able to keep track of this set. This is done in two steps: first, identifying numbers are assigned to each Category 1 parameter as shown in Table 3-I below.

Table 3-I. Category 1 Identifiers

Parameter Type	Symbol	Variable
1	α	Right Ascension
2	δ	Declination
3	β	Flight Path Angle
4	A	Azimuth of Velocity Vector
5	R	Radius
6	v	Velocity
7	$C_D A/2m$	Drag Parameter
8	K	Drag Parameter Variation

Second, the IVSTR (NIDP) array consists of identifing numbers for the Category 1 variables to be solved. For example, if an analyst requested α, δ, β, A, R, v to be solved for, the IVSTR (NIDP) array would be:

IVSTR (NIDP) = 1
IVSTR (NIDP+1) = 2
IVSTR (NIDP+2) = 3
IVSTR (NIDP+3) = 4
IVSTR (NIDP+4) = 5
IVSTR (NIDP+5) = 6

3.2.2 Category 2 Variables

Category 2 variables are composed of the following ten parameters: R_b, A_b, E_R, $\dot{R}_b$, HA_b, D_b, t_b, ϕ_b, λ_b, h_b. Category 2 parameters differ

from Category 1 parameters in that they are sensor dependent. Through the CAT2 input cards the analyst is permitted to select, by sensor number, those Category 2 parameters to be included in the differential correction process. The fact that Category 2 variables are sensor dependent creates a bookkeeping problem for the program. It becomes necessary to know what type of Category 2 parameter is being solved, what sensor is involved, and where this parameter will be located in the solution vector. (All Category 2 variables come after Category 1 variables in the solution vector). This bookkeeping is accomplished through two arrays in variable storage and a single word (called code word) found in the master sensor table (one for each sensor in the table). The format and use of these arrays and code words are described below. First, identifying numbers are assigned to each of the Category 2 variables as shown in Table 3-II below:

Table 3-II. Category 2 Identifiers

Bias Type	Symbol	Variable
1	R_b	Range bias
2	A_b	Azimuth bias
3	E_b	Elevation bias
4	$\dot{R}_b$	Range rate bias
5	HA_b	Hour angle bias
6	D_b	Declination bias
7	t_b	Time bias
8	ϕ_b	Sensor latitude bias
9	λ_b	Sensor longitude bias
10	h_b	Sensor altitude bias

Each element of the IVSTR (NPRCD) array in fixed point variable storage contains two pieces of information about a Category 2 variable:

- What type of bias it is (T)
- What place it occupies in the solution vector (P)

This information is contained in the IVSTR (NPRCD) array as a single integer of the form

$$T * 100 + P$$

The code word given in the master sensor table for each sensor tells where to look in the IVSTR (NPRCD) array for additional information concerning Category 2 variables being solved for the sensor. If the code word of a sensor equals zero, then no Category 2 variables are being considered for the sensor. If the code word of a sensor is nonzero, it has the following form:

$$A * 100 + B$$

A and B refer to the starting and stopping points in the IVSTR (NPRCD) array where the program can find the numbers identifying Category 2 variables which are being solved for this sensor.

Finally, the VSTR (NPBIS) array contains in floating point form, the current estimates of the Category 2 variables defined in the IVSTR (NPRCD) array. Thus, A and B can also be used to locate the current estimates of Category 2 variables of a sensor. The VSTR (NPBIS) array is updated after each iteration of the differential correction.

3.2.3 Examples

Tables 3-III and 3-IV were prepared assuming that observations from three sensors, S1, S2, and S3, were being used in a differential correction, and that the solution for Category 1 variables α, δ, β, A, R, v, and Category 2 variables S2-R_b, A_b, E_b, and S3-t_b, ϕ_b, λ_b, h_b was desired. Table 3-III gives the form of the solution vector and Table 3-IV gives the contents of the IVSTR (NIDP), IVSTR (NPRCD), and VSTR (NPBIS) arrays.

3.3 OBSERVATION NUMBERING SYSTEM

ESPOD programs maintain an internal numbering system for the observations being processed. This system becomes particularly useful in the computations of partial derivatives of observations with respect to

Table 3-III. Example of Solution Vector

Variable Place in Solution Vector	Symbol	Variable Name
1	α	Right ascension
2	δ	Declination
3	β	Flight path angle
4	A	Azimuth of velocity vector
5	R	Radius
6	v	Velocity
7	S2, R_b	Range bias of sensor S2
8	S2, A_b	Azimuth bias of sensor S2
9	S2, E_b	Elevation bias of sensor S2
10	S3, t_b	Time bias of sensor S3
11	S3, ϕ_b	Latitude bias of sensor S3
12	S3, λ_b	Longitude bias of sensor S3
13	S3, h_b	Altitude bias of sensor S3

Table 3-IV. Example of Internal Storage, Defining Solution Vector

Array Element No.	Contents of IVSTR(NIDP)	Contents of IVSTR(NPRCD)	Contents of VSTR(NPBIS)
1	1	107	Current estimate of S2, R_b
2	2	208	Current estimate of S2, A_b
3	3	309	Current estimate of S2, E_b
4	4	710	Current estimate of S3, t_b
5	5	811	Current estimate of S3, ϕ_b
6	6	912	Current estimate of S3, λ_b
7		1013	Current estimate of S3, h_b

S1 code word = 0
S2 code word = 103
S3 code word = 407

parameters in the solution vector. (See subroutine DRDP and PUPB). The numbering system is:

Type	Observation
1	Range
2	Azimuth
3	Elevation
4	Range rate
5	Hour angle
6	Declination

3.4 UNITS OF ESPOD PARAMETERS

The internal and external dimensional units are given in Table 3-V.

Table 3-V. ESPOD Units

Quantity	Internal Units	External Units
Distance	Earth radii (e.r.)	Kilometer
Velocity	Earth radii per minute	Kilometer per second
Angular Measure	Radians	Degrees
Area	$(\text{Meters})^2$	$(\text{Meters})^2$
Mass	Kilogram	Kilogram

3.5 ALLOTMENT OF CORE STORAGE

3.5.1 Map of Core

	OCTAL
	0
EXECMØDI and EXECMØD2	
	15524
Input Buffer Storage Constants Blocks Initial Input Block Variable Storage Assignment Numbers Buffer Block for Intermediate Tape Handling Temporary Storage ESPØDDC and ESPØDEPH Working Storage Variable Storage (Fixed and Floating Point)	
	33661
ESPØD Driver	
	34161
ESPØD ESPØDDC ESPØDEPH	
	63234
BMEWS	

Figure 3-1. Map of Core

3.5.2 Variables and Arrays in Storage

The following list includes:

- Name of item
- Relative location from CØMMØN storage base address YYYY
- Dimension if item is an array
- Definition of item

The list is ordered sequentially by relative location.

Name	2000	DIM	Definition of Variable or Array
CARBUF	1	128	Input buffer
CØNVR	129		Conditional start flag
CDRUNB	130		Input buffer counter
FIRSTFL	131		Flag to indicate "first case"
CWE	141		Earth's rotational rate (real 1 min)
CELLIP	142		Ellipticity of the Earth
CGNØM	143		Gravitational constant
CMU	144		GM Earth (e.r.^{3}/min^2)
CGMR	145	7	GM ratios (E, M, S, V, M, E-M, J)
BFLAGS	152	7	Flags to indicate bodies to be considered
CJ	159	11	$J_2, J_3, J_4, \ldots, J_{12}$
CJNM	170	36	6 x 6 array containing J_m below diagonal, $J_{n,n}$ on the diagonal and $\lambda_{n,m}$ above the diagonal
CLAMNN	206	5	$\lambda_{n,n}$, n = 2, 6
CDAD2M	211		$C_D A/2m$
CK	212		K for drag variation
CKSLCT	213		Selector for K, = 0., 1., or 2.

Name	2000	DIM	Definition of Variable or Array
CDRAGM	214		Selects atmosphere model to be used
CFTER	215		Conversion from Earth radii to ft
CKMFT	216		Conversion from ft to km
CKMER	217		Conversion from Earth radii to km
CDTER	218		Conversion from Earth radii to km
CMTER	219		Conversion from Earth radii to meters
CERAU	220		Conversion from A. U. to Earth radii
CDEG	221		Conversion from radians to degrees
CFTNM	222		Conversion from n mi to ft
CNMER	223		Conversion from Earth radii to n mi
CVTERM	224		Conversion from e. r./min to km/sec
CSIG	225	120	60 sets of sensor sigmas (packed)
CDAYMN	345	12	Number days in month
CAPF10	357	91	30 sets t, A_p, F_{10}
CPI	448		π
C2PI	449		2π
CAE	450		a_e
CBE	451		b_e
CKRMS	452		N for N(RMS) deletion
CØMLST	453		Dimension of CØMMØN
CFTEPS	454		ϵ for convergence criterion
CJD50	455		Julian date January 0, 1950
CBØUND	456	18	Nominal set of bounds

Name	2000	DIM	Definition of Variable or Array
KØUT	474		Output tape number
IØUT	475		
KIN	476		Input tape number
MT	477		Observation tape number
NØUT	478		Ephemeris tape number
CHMAX	479		Maximum step size
CHMIN	480		Minimum step size
CYMIN	481		Parameter for variable step integration
CER	482		Parameter for variable step integration
NRRR	483		Ratio of Cowell to Runge-Kutta step size
TSTEP	484		Nominal step size
CSTYPE	485	120	Sensor types for σ, $\overline{N}_s$ and N
CP	605	16	4 x 4 array of polynomial coeff. for refraction
CLDSTR	641		Cold-start, non-cold-start flag
TEPØCH	642		Epoch time, min from midnight
TJDATE	643		Julian date of midnight, epoch day
DYEAR	644		Epoch year
DMNTH	645		Epoch month
DDAY	646		Epoch day
DHØUR	647		Epoch hour
DMIN	648		Epoch min
DSEC	649		Epoch sec
DTYPE	650		Initial conditions type
TALFAG	651		α_g for midnight day of epoch
DNUT	652		Nutation correction

Name	2000	DIM	Definition of Variable or Array
DSDAY	653		Epoch day, days from beginning of year
DSFDAY	654		Epoch time, fraction of day
DBASE	655		Number days from 1950 to day of epoch
DNREV	656	6	Control cells for seven-card input
DELTT	662	17	Sets of Δt, t
DVEHN	679	3	Vehicle number and name (BCD)
DHEAD	682	2	Header from JDC card
PREFLG	684	10	Preprocessor control flags
DCFLG	694	10	DC package control flags
PSTFLG	704	10	Post-processor control flags
DTARG	714	20	Temporary location for DAC or PRDCT card image
SEQ	734		Sequence number from DAC cards
DFL	735		Flag for dynamic atmosphere
HEADER	736	8	Contents of REM card
FGITIM	744		Flag to indicate ITIME card read
FGICØN	745		Flag to indicate ICØND card read
FGICTY	746		Flag to indicate ICTYPE card read
FGELEM	747		Flag to indicate element cards read
FGCAT1	748		Flag to indicate CAT1 card read
FGCAT2	749		Flag to indicate CAT2 cards read
FGBNDS	750		Flag to indicate BNDS cards read
FGDELE	751		Flag to indicate DELETE cards read
NØEPØC	752		Flag to indicate epoch not established
DLPSI	753		$\Delta\psi$

Name	2000	DIM	Definition of Variable or Array
DLEPS	754		$\Delta\epsilon$
SNEPS	755		$\sin \epsilon$
CSEPS	756		$\cos \epsilon$
DTMAX	757		t_{max} for DC to check for bad observation times
TNØMX	758	6	Initial Cartesian coordinates $(x, y, z, \dot{x}, \dot{y}, \dot{z})$
TNØMP	764	6	Initial spherical coordinates $(\alpha, \delta, \beta, A, R, v)$
TMNEL	770	10	Initial seven-card element sets
TCLSEL	780	8	Classical elements $(a, e, i, \Omega, \omega, M)$
ZØNAL	788	11	Flags for zonal harmonics
SECT	799	5	Flags for sectoral harmonics
TESS	804	14	Flags for tesseral harmonics
DAREA	818		Area of spacecraft M^2
DMASS	819		Mass of spacecraft kg
CSOLC	820		Solar Constant S watts/m^2
CLIGHT	821		Speed of light e.r./min
FGAUX	822		= 0 No SYTPES in DBUFS = N No sets of STYPE entries in DBUFS
NPR	850		Total number of parameters to solve for
NDPR	851		Total number of CAT1 variables to solve for
NICPR	852		Total number of spherical coordinates to solve for
NITER	853		Maximum number of iterations
NMBER	854		Number of observations

Name	2000	DIM	Definition of Variable or Array
NDTCT	855		Counter for Δt, t table
NITCT	856		Iteration counter
NIDENT	857		Number of entries in NIDLED list
N1	858		Counters for geopotential routine
N2	859		
N3	860		
FLVE	861		Parameters for numerical integration
SKIP	862		
NIDP	863		Identifiers for starting locations of arrays in VSTR and IVSTR
NPRCD	864		
NPBIS	865		
NARØW	866		
NBDNS	867		
NPAR	868		
NDPAR1	869		
NDPAR2	870		
NDPAR3	871		
NDPAR4	872		
NSCALE	873		
NATA	874		
NR	875		
NIDLED	876		
NRTMP	877		
NSSTB	878		
NSMAT	879		

Name	2000	DIM	Definition of Variable or Array
NSTAT	880		Identifiers for starting locations of arrays in VSTR and IVSTR
NUBS	881		
DBUFS	882	256	Auxiliary buffer storage
DTMP	1138	100	Saves station number and code word for those stations with code word $\neq 0$
DATA	1438	1260	Input storage
TEMP	1238	200	Temporary storage
TRAJX	1438	57	Temporary location for output from TRAJ subroutine-stores $x, y, z, \dot{x}, \dot{y}, \dot{z}$
TLIST	1495	490	Numerical integration working storage
TICRT	1985	6	Nominal Cartesian coordinates
TIPØL	1991	6	Nominal spherical coordinates
TG	1997		Time to integrate to
TSUS	1998		Current total SØS
TSUSP	1999	4	Predicted SØS for next iteration
TSUSB	2003		Best SØS so far
TMBIS	2004		Current estimate of time bias for the observation time being considered (if applicable)
TMINUS	2005		Flag to indicate integration times before epoch
IFTEX	2006		Indicates mode of exit from FIT = 1 Converged = 2 Max iterations and converging = 3 Failed K*BNDS/8 = 4 Normal return = 5 Max iterations and diverging

Name	2000	DIM	Definition of Variable or Array
TUBSEF	2007		EØF flag for reading observations
TRHØA	2008		Density, kilograms/m^3
TALT	2009		Altitude in meters
TDPDX	2010	64	Contains matrices of partials for covariance matrix update
TRS	2074		Distance E→S
TRM	2075		Distance E→M
TZ	2076		Indicates if solution was affected by bounds
XBSQ	2077		Scale factor for BNDS to cause subsequent solutions to be affected by bounds
PHIH	2078	70	Tables for PAETZOLD dynamic atmosphere
THETH	2148	70	
ALT	2218	70	
PSTAR	2288	70	
TDRAG	2358	3	Three components of acceleration due to drag
TV	2361	3	Three components of Earth-fixed velocity
TVA	2364		Magnitude of Earth-fixed velocity
TR	2365		R
TR2	2366		R^2
TR3	2367		R^3
TR5	2368		R^5
TR7	2369		R^7
TDØN	2370		Flag for drag model
TPØT	2371	3	Total acceleration due to Earth's potential field
CØLA	2374		$\cos\phi$ ϕ = latitude

Name	2000	DIM	Definition of Variable or Array
SILA	2375		sin ϕ
SIPH	2376		sin λ λ = longitude
CØPH	2377		cos λ
SNALF	2378		sin α α = right ascension
CSALF	2379		cos α
FJ	2380	12	Working storage for generalized geopotential subroutine
C	2392	36 (6 x 6)	
S	2428	36 (6 x 6)	
CØUNT	2464		Lines counter
CAP	2465		Working values of A_p and F_{10}
CF10	2466		
XN	2467	21	Position of bodies table (from ephemeris)
XNDØT	2488	21	Velocities of bodies table (from ephemeris)
RJUPT	2509		Jupiter inclusion radius
TSEC	2510	2	Interpolation time (sec from 1950)
INTRX	2512		Exit flag from INTR
CENTER	2513		Central body number for INTR
TALFA	2514		α(e.r.^{2}/min)
TRPRES	2515	3	$\ddot{x}$, $\ddot{y}$, $\ddot{z}$, due to radiation pressure
TBPERT	2518	3	$\ddot{x}$, $\ddot{y}$, $\ddot{z}$, due to bodies
TCRASH	2521		$\neq 0$ Vehicle below 1 e.r. $= 0$ okay
PMAT	2522	9 (3 x 3)	Matrix used in computation of variational equation second derivatives

Name	2000	DIM	Definition of Variable or Array
VMAT	2531	9(3x3)	Matrix used in computation of variational equation second derivatives
PUBS	2550	8	Sensor number, time, R, A, E, $\dot{R}$, α, δ table
PSTAT	2558	12	Working storage for sensor information
PCSALF	2570		$\cos(\alpha_g)$ $\quad \alpha_g = \alpha_{go} + \lambda + \omega_e t$
PSNALF	2571		$\sin(\alpha_g)$
PWI	2572	3	Vector (w_1, w_2, w_3)
PWDTI	2575	3	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
PUI	2578	3	Vector (u_1, u_2, u_3)
PVI	2581	3	Vector (V_1, V_2, V_3)
PV	2584		$\sqrt{V_1^2 + V_2^2}$
PRSUB1	2585		$R_1 = V_R$
PSNE	2586		$\sin E_c$
PCSE	2587		$\cos E_c$
PSNA	2588		$\sin A_c$
PCSA	2589		$\cos A_c$
PCMR	2590		R = computed slant range
PWPP	2591	24	Partial derivatives
PWDTPP	2615	24	Partial derivatives
PRESD	2639	6	Residuals (measured-computed)
IPFRST	2645		0 to indicate first time in RADR
PLSTSN	2646		Number of last sensor processed by RADR
PUDTI	2647	3	Vector $(\dot{u}_1, \dot{u}_2, \dot{u}_3)$
PSIG	2650	6	Sigma list

Name	2000	DIM	Definition of Variable or Array
PØBCNT	2656		Total number of accepted observations
IRCNT	2657	6	Cells for partials print
PDELFG	2663	6	Cells for partials print
PRESDT	2669	11	Cells for partials print
PKSUBS	2680		Rejection criterion scale factor
VSTR	2700	4700	Floating point variable storage
IVSTR	2700	4700	Fixed point variable storage
YYYY		7400	Base address for working storage within common
YYZZ		13745	CØMMØN

3.6 DESCRIPTION OF DIMENSIONAL ARRAYS

The following sections give further detailed information regarding the important arrays stored in core. They are ordered alphabetically by the name of the array. The identification line gives the:

a) Name of the array

b) Dimension of the array

c) Relative location from common storage base address

3.6.1 ALT (70), YYYY (2218)

Altitudes at selected increments are stored in ALT. This table is then used to generate the Paetzold dynamic atmosphere tables THETH, PHIH and PSTAR.

$$ALT(I) = 130. + 10\ (I-1) \qquad \text{for } I = 1, \cdots, 28$$

$$ALT(I) = 420. + 20\ (I-29) \qquad \text{for } I = 29, \cdots, 58$$

$$ALT(I) = 1050. + 50\ (I-59) \qquad \text{for } I = 59, \cdots, 70$$

The altitudes are in kilometers.

3.6.2 C (6 x 6) YYYY (2392)

C is formed from the tesseral or sectorial harmonics requested by the arrays TESS or SECT respectively.

$$C_{n,m} = J_{n,m} \cos m\lambda_{n,m}$$

3.6.3 FJ (12), YYYY (2380)

For every nonzero entry in the ZØNAL array the corresponding zonal harmonic is transferred to FJ.

$$FJ(I) = J_I \; ; \; I = 2, \cdot \cdot \cdot , 12$$

3.6.4 IRCNT (6), YYYY (2657)

A count is kept of the six possible residuals computed for each station. During the residuals print the first nonzero element of IRCNT is printed to identify the first residual of each line. If IRCNT(I) = 0, the I^{th} residual of a possible six has not been considered.

3.6.5 PDELFG (6), YYYY (2668)

Corresponding to each printed residual of R, A, E, $\dot{R}$, α, or δ a BCD character is stored in PDELFG to signal deletion if any.

PDELFG(I) = blank	No deletion
PDELFG(I) = *	Deletion by input number
PDELFG(I) = G	Deletion (gross outlier)
PDELFG(I) = K	Deletion (K RMS)

3.6.6 PHIH (70), YYYY (2078)

For each of the 70 altitudes stored in ALT a corresponding value of PHIH is computed. The PHIH table is then used to interpolate values of the Paetzold angle $\psi(h)$. A typical entry of PHIH is computed from

$$\psi(h) = i(h) \frac{(220 - F)}{F} \theta_i(h)$$

where $i(h)$ and $\theta_i(h)$ are polynomial approximations in altitude.

3.6.7 PRESD (6), YYYY (2639)

The residuals (measured-completed) applicable to a given sensor are stored in PRESD.

PRESD(1) = ΔR (e.r.)

(2) = ΔA (radians)

(3) = ΔE (radians)

(4) = $\Delta \dot{R}$ (e.r./min)

(5) = $\Delta \alpha$ (radians)

(6) = $\Delta \delta$ (radians)

3.6.8 PRESDT (11), YYYY (2669)

For output purposes the residuals are stored in PRESDT.

PRESDT(1) = ΔR (e.r.)

(2) = ΔA, Δ HA

(3) = ΔE, ΔDEC

(4) = $\Delta \dot{R}$ (e.r./min)

(5) = Δu, ΔS, $\Delta \phi$ (option dependent)

(6) = Δv, ΔT, $\Delta \lambda$ (option dependent)

(7) = Δw, Δw, Δh (option dependent)

(8) = VMAG (e.r.)

(9) = ΔT (min)

(10) = u (rad)

(11) = β (rad)

3.6.9 PSIG (6), YYYY (2650)

For each station the appropriate set of sigmas is moved from the CSIG table, scaled and stored in PSIG.

$$
\begin{aligned}
\text{PSIG}(1) &= \sigma_R \text{ (e. r.)} \\
(2) &= \sigma_A \text{ (radians)} \\
(3) &= \sigma_E \text{ (radians)} \\
(4) &= \sigma_{\dot{R}} \text{ (e. r. /min)} \\
(5) &= \sigma_\alpha \text{ (radians)} \\
(6) &= \sigma_\delta \text{ (radians)}
\end{aligned}
$$

3.6. 10 PSTAR (70), YYYY (2288)

For each of the 70 altitudes stored in ALT a corresponding value of PSTAR is computed. The PSTAR table is then used for atmospheric density interpolations. A typical entry of PSTAR is computed from

$$
\log \rho^* = \log \rho_s(h) - i(h) \frac{220 - F}{F} + K(h) \frac{A_p}{200} - K(h)\, m
$$

where $m = g(a) + (220 - F)\left[0.006 - 0.002\, g(a)\right]$ and $\log \rho_s(h)$, $i(h)$ and $K(h)$ are computed from polynomial approximations in altitude. $g(a)$ is a polynomial seasonal dependent.

3.6.11 PSTAT (12), YYYY (2558)

Sensor information for a particular station is moved from the Master Sensor Table to PSTAT.

PSTAT(1) = ϕ Latitude

(2) = λ Longitude

(3) = h Altitude

(4) = $\cos \phi$

(5) = $\sin \phi$

(6) = $\alpha_{go} + \lambda$

(7) = w_1^s

(8) = w_3^s

(9) = code word

(10)

.

. not used

.

(12)

If the code word is zero, Category 2 variables are not being considered for this station. If nonzero, the word contains information for locating the current biases to be applied to this station.

3.6.12 PUBS (8), YYYY (2250)

For a specified time and station, observations are moved from variable storage to PUBS.

PUBS(1) = sensor number

(2) = time

(3) = R

(4) = A

(5) = E

(6) = $\dot{R}$

(7) = α

(8) = δ

3.6.13 PUDTI (3), YYYY (2467)

The topocentric direction cosines of velocity vector in horizon system are stored in PUDTI.

$$PUDTI(1) = \dot{u}_1$$
$$(2) = \dot{u}_2$$
$$(3) = \dot{u}_3$$

3.6.14 PUI (3)

Topocentric direction cosines for the vehicle position in horizon system are stored in PUI.

$$PUI(1) = u_1$$
$$(2) = u_2$$
$$(3) = u_3$$

3.6.15 PVI (3), YYYY (2581)

Topocentric direction cosines of vehicle in horizon system are stored in PVI.

$$PVI(1) = v_1$$
$$(2) = v_2$$
$$(3) = v_3$$

3.6.16 PWDTI (3), YYYY (3)

Geocentric velocity of vehicle is stored in PWDTI.

$$PWDTI(1) = \dot{w}_1$$
$$(2) = \dot{w}_2$$
$$(3) = \dot{w}_3$$

3.6.17 PWDTPP (24), YYYY (2615)

The variational equations in velocity are rotated to meridian coordinates from TRAJX storage and stored in PWDTPP.

$$\text{PWDTPP}\left[1 + 3(i - 1)\right], \ldots, \left[3 + 3(i - 1)\right] = \left(\frac{\partial \dot{w}_1}{\partial p_i}, \frac{\partial \dot{w}_2}{\partial p_i}, \frac{\partial \dot{w}_3}{\partial p_i}\right)$$

$i = 1, \cdots, n$

n is the number of parameters p_i to be solved for from the list (α_o, δ_o, β_o, ρ_o, r_o, v_o, $C_D A/2m$, K).

3.6.18 PWI (3), YYYY (2572)

Geocentric position of vehicle is stored in PWI.

$$\begin{aligned} \text{PWI}(1) &= w_1 \\ (2) &= w_2 \\ (3) &= w_3 \end{aligned}$$

3.6.19 PWPP (24), YYYY (2591)

The variational equations are rotated to meridian coordinates from TRAJX storage and stored in PWPP.

$$\text{PWPP}\left[1 + 3(i - 1)\right], \ldots, \left[3 + 3(i - 1)\right] = \left(\frac{\partial w_1}{\partial p_i}, \frac{\partial w_2}{\partial p_i}, \frac{\partial w_3}{\partial p_i}\right)$$

$i = 1, \cdots, n$

n is the number of parameters p_i to be solved for from the list (α_o, δ_o, β_o, A_o, r_o, v_o, $C_D A/2m$, K).

3.6.20 S (6 x 6), YYYY (2428)

S is formed from the tesseral or sectorial harmonics requested by the arrays TESS or SECT respectively.

$$S_{n,m} = J_{n,m} \sin m\lambda_{n,m}$$

3.6.21 SECT (5), YYYY (799)

SECT is tested for the inclusion of sectorial harmonics.

If SECT(I) $\neq 0$, then $\lambda_{I,I}$ and $J_{I,I}$ are included in Earth potential model.

3.6.22 TBPERT (3), YYYY (2518)

The total acceleration on the vehicle due to selected bodies is stored in TBPERT.

$$\begin{aligned} \text{TBPERT}(1) &= \ddot{x}\ (\text{e.r.}/\text{min}^2) \\ (2) &= \ddot{y}\ (\text{e.r.}/\text{min}^2) \\ (3) &= \ddot{z}\ (\text{e.r.}/\text{min}^2) \end{aligned}$$

3.6.23 TDRAG (3), YYYY (2358)

The components of acceleration due to drag are stored in TDRAG(3).

$$\begin{aligned} \text{TDRAG}(1) &= \ddot{x}_{drag}\ (\text{e.r.}/\text{min}^2) \\ (2) &= \ddot{y}_{drag}\ (\text{e.r.}/\text{min}^2) \\ (3) &= \ddot{z}_{drag}\ (\text{e.r.}/\text{min}^2) \end{aligned}$$

3.6.24 TESS (14), YYYY (804)

Up to 14 tesseral harmonics may be specified in the array TESS.

If $\lambda_{n,m}$ and $J_{n,m}$ are to be included in the Earth potential model, then TESS(I) = 10 n + m.

3.6.25 THETH (70), YYYY (2148)

For each of the 70 altitudes stored in ALT a corresponding value of THETH is computed. The THETH table is then used to interpolate values of the Paetzold angle $\theta(h)$. A typical entry of THETH is computed from

$$\theta(h) = \theta_s(h) - \Delta_1(h) \frac{i(h)\left(\frac{220 - F}{F}\right) + m(h)}{i(h) + a(h)} - \Delta_2\theta(h)\left(\frac{200 - F}{F}\right)$$

where $\theta_s(h)$, $\Delta_1(h)$, $\Delta_2\theta(h)$, $i(h)$ and $a(h)$ are computed from polynomial approximations in altitude.

$$m(h) = g(a) + (200 - F)\left[0.006 - 0.002\ g(a)\right]$$

where g(a) is a polynomial seasonal dependent.

3.6.26 TICRT (6), YYYY (1985)

The Cartesian coordinates of the current solution vector are stored in TICRT for each iteration.

TICRT(1) = x (e.r.)

(2) = y (e.r.)

(3) = z (e.r.)

(4) = $\dot{x}$ (e.r./min)

(5) = $\dot{y}$ (e.r./min)

(6) = $\dot{z}$ (e.r./min)

3.6.27 TIPØL (6), YYYY (1991)

The polar spherical coordinates of the current solution vector are stored in TIPØL for each iteration.

TIPØL(1) = α (radians)

(2) = δ (radians)

(3) = β (radians)

(4) = A (radians)

(5) = R (e.r.)

(6) = v (e.r./min)

3.6.28 TLIST (490), YYYY (1495)

See description of subroutine TRAJ.

3.6.29 TMNEL (10), YYYY (770)

The array TMNEL is computed from SPADATS element input.

TMNEL(1)	$= N_o$	Epoch revolution number
(2)	$= a_o$	Semimajor axis in e.r.
(3)	$= e_o$	Eccentricity
(4)	$= i$	Inclination in radians
(5)	$= \Omega_o$	R.A. or ascending node in degrees
(6)	$= \omega_o$	Argument of perigee in degrees
(7)	$= L_o$	Mean longitude in degrees
(8)	$= C_o$	Rate of change of anomilistic period in days/rev^2
(9)	$= P_N$	Nodal period in days/rev
(10)	$= C_N$	Rate of change of nodal period in days/rev^2

3.6.30 TNØMP (6), YYYY (764)

TNØMP contains the initial estimates of the polar elements.

TNØMP(1) = α (deg)

(2) = δ (deg)

(3) = β (deg)

(4) = A (deg)

(5) = R (km)

(6) = v (km/sec)

The differential correction package updates TNØMP each iteration.

3.6.31 TNØMX (6), YYYY (758)

TNØMX contains the initial estimates of Cartesian position and velocity.

$$\begin{aligned} \text{TNØMX}(1) &= x \text{ (km)} \\ (2) &= y \text{ (km)} \\ (3) &= z \text{ (km)} \\ (4) &= \dot{x} \text{ (km/sec)} \\ (5) &= \dot{y} \text{ (km/sec)} \\ (6) &= \dot{z} \text{ (km/sec)} \end{aligned}$$

The differential correction package updates TNØMX each iteration.

3.6.32 TPØT (3), YYYY (2371)

The components of total acceleration due to Earth's potential field are stored in TPØT.

$$\begin{aligned} \text{TPØT}(1) &= \ddot{x} \text{ (e.r./min}^2) \\ (2) &= \ddot{y} \text{ (e.r./min}^2) \\ (3) &= \ddot{z} \text{ (e.r./min}^2) \end{aligned}$$

3.6.33 TRAJX (57), YYYY (1438)

For the current integration time TRAJX contains the position, velocity and acceleration vectors of the vehicle. If variational equations have been integrated they are also present in TRAJX.

$$\begin{aligned} \text{TRAJX}(1), \cdots (3) &= (x, y, z) \\ (4), \cdots (6) &= (\dot{x}, \dot{y}, \dot{z}) \\ (7), \cdots (9) &= (\ddot{x}, \ddot{y}, \ddot{z}) \end{aligned}$$

$$\text{TRAJX}\left[10 + 6(i-1)\right], \cdots \left[12 + 6(i-1)\right] = \frac{\partial x}{\partial p_i}, \frac{\partial y}{\partial p_i}, \frac{\partial z}{\partial p_i}$$

$$i = 1, \cdots, n$$

$$\text{TRAJX}\left[13 + 6(i-1)\right], \cdots \left[15 + 6(i-1)\right] = \frac{\partial \dot{x}}{\partial p_i}, \frac{\partial \dot{y}}{\partial p_i}, \frac{\partial \dot{z}}{\partial p_i}$$

n is the number of parameters p_i to be solved for from the list $(\alpha_o, \delta_o, \beta_o, A_o, R_o, v_o, C_D A/2m, K)$.

3.6.34 TRPRES (3), YYYY (2515)

The acceleration on the vehicle due to the sun's radiation pressure is stored in TRPRES.

$$\begin{aligned} \text{TRPRES}(1) &= \ddot{x}_{rad} \ (\text{e.r.}/\text{min}^2) \\ (2) &= \ddot{y}_{rad} \ (\text{e.r.}/\text{min}^2) \\ (3) &= \ddot{z}_{rad} \ (\text{e.r.}/\text{min}^2) \end{aligned}$$

3.6.35 TSUSP (4), YYYY (1999)

TSUSP contains the predicted RMS corresponding to the four candidate solutions proposed by the least squares procedure.

TSUSP(1)	Predicted RMS for nominal bounds
TSUSP(2)	Predicted RMS for K* (nominal bounds/2)
TSUSP(3)	Predicted RMS for K* (nominal bounds/4)
TSUSP(4)	Predicted RMS for K* (nominal bounds/8)

3.6.36 TV (3), YYYY (2361)

The components of velocity of the vehicle relative to the atmosphere are stored in TV.

$$\begin{aligned} \text{TV}(1) &= \dot{x}_A \ (\text{e.r.}/\text{min}) \\ (2) &= \dot{y}_A \ (\text{e.r.}/\text{min}) \\ (3) &= \dot{z}_A \ (\text{e.r.}/\text{min}) \end{aligned}$$

3.6.37 VSTR (6300), IVSTR (6300), YYYY (2700)

See section Variable Storage

3.6.38 <u>XN (21), YYYY (2467)</u>

XN contains the position vectors for up to seven bodies.

XN(1) , $\cdots$, (3) = (x, y, z)(Earth)

XN(4) , $\cdots$, (6) = (x, y, z)(moon)

XN(7) , $\cdots$, (9) = (x, y, z)(sun)

XN(10), $\cdots$, (12) = (x, y, z)(Venus)

XN(13), $\cdots$, (15) = (x, y, z)(Mars)

XN(16), $\cdots$, (18) = (x, y, z)(Saturn)

XN(19), $\cdots$, (21) = (x, y, z)(Jupiter)

The coordinates are interpolated from an ephemeris tape.

3.6.39 <u>XNDØT (21), YYYY (2488)</u>

XNDØT contains the velocity vectors for up to seven bodies.

XNDØT(1) , $\cdots$, (3) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) Earth

XNDØT(4) , $\cdots$, (6) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) moon

XNDØT(7) , $\cdots$, (9) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) sun

XNDØT(10), $\cdots$, (12) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) Venus

XNDØT(13), $\cdots$, (15) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) Mars

XNDØT(16), $\cdots$, (18) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) Saturn

XNDØT(19), $\cdots$, (21) = ($\dot{x}$, $\dot{y}$, $\dot{z}$) Jupiter

The coordinates are interpolated from an ephemeris tape.

3.6.40 <u>ZØNAL (11), YYYY (788)</u>

ZØNAL is tested for the inclusion of zonal harmonics. If ZØNAL (I) $\neq$ 0, then J_{I+1} is included in Earth potential model.

3.7 MAGNETIC TAPE FORMATS

ESPOD utilizes magnetic tapes in numerous formats. These formats are described by the sections which follow as listed below.

3.7.1 Tape Setup and Description

Table 3-VI shows how ESPOD interfaces with magnetic tapes. The following codes are used:

Write ring required

May be left out if proper conditions are met

Table 3-VI. Program Tapes

Logical Tape No.	Setup	Tape Description
0	Scratch	Data is transferred to this tape.
1	System	RPL library of SPS (Semiautomatic Programming System) programs.
2	Schedule	Job tape (input).
3	SEAI	Backup tape for logical tape No. 4.
4	SEAI	Master SEAI (sensor, elements, acquisition and information files) tape. The ESPØD program uses only the sensors and elements off this tape.
5		Not used.
6	SRADU	SRADU tape contains observations existing prior to the run.
7	Scratch	The ESPØD program writes blocks of common data and observations just processed on this tape. (70 TAPE7)
8		Planetary ephemeris tape.
9		Not used.
10	Scratch	Trajectory tape (optional).
11	Output	Off-line output tape.

3.7.2 SEAI Tape Format

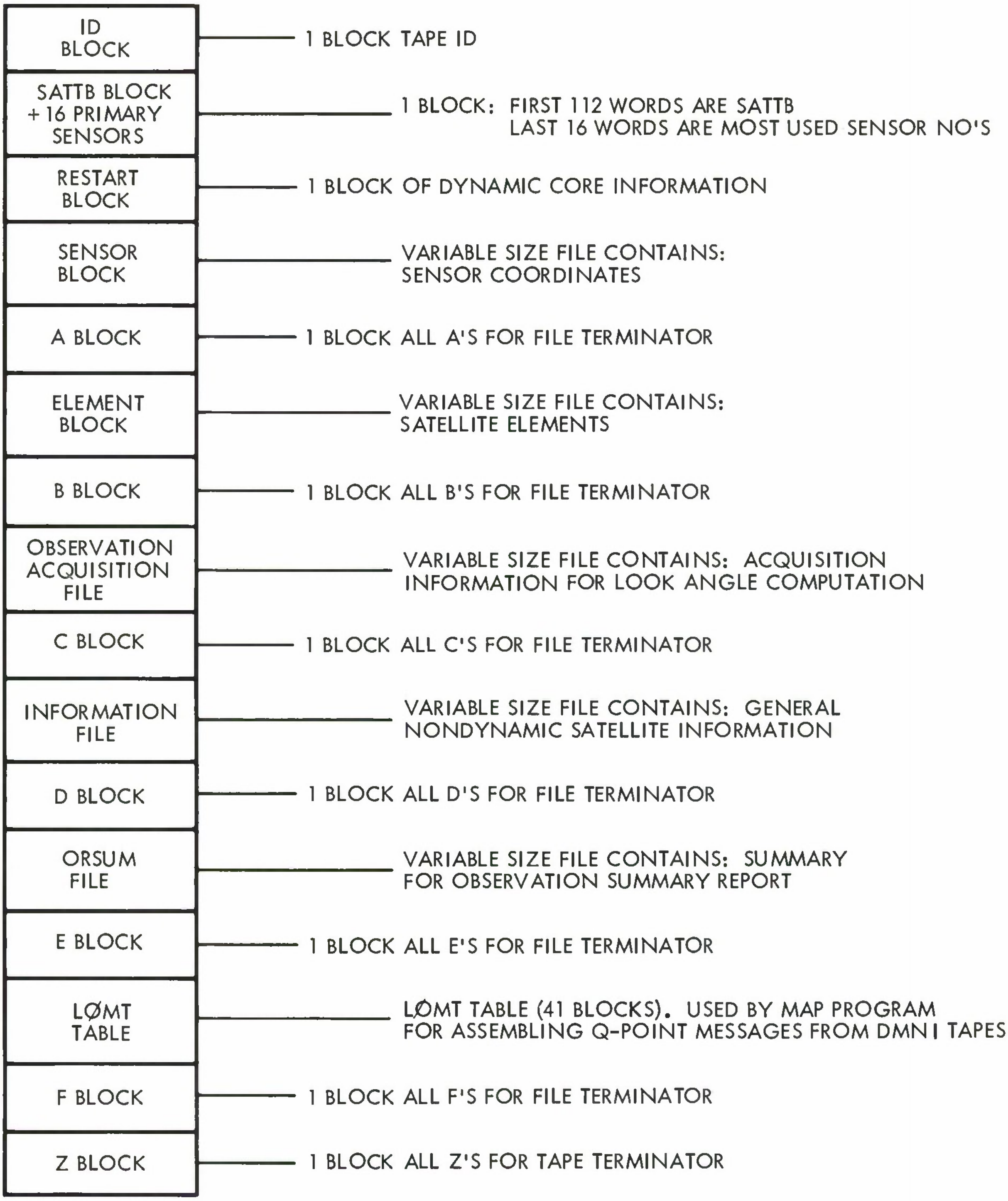
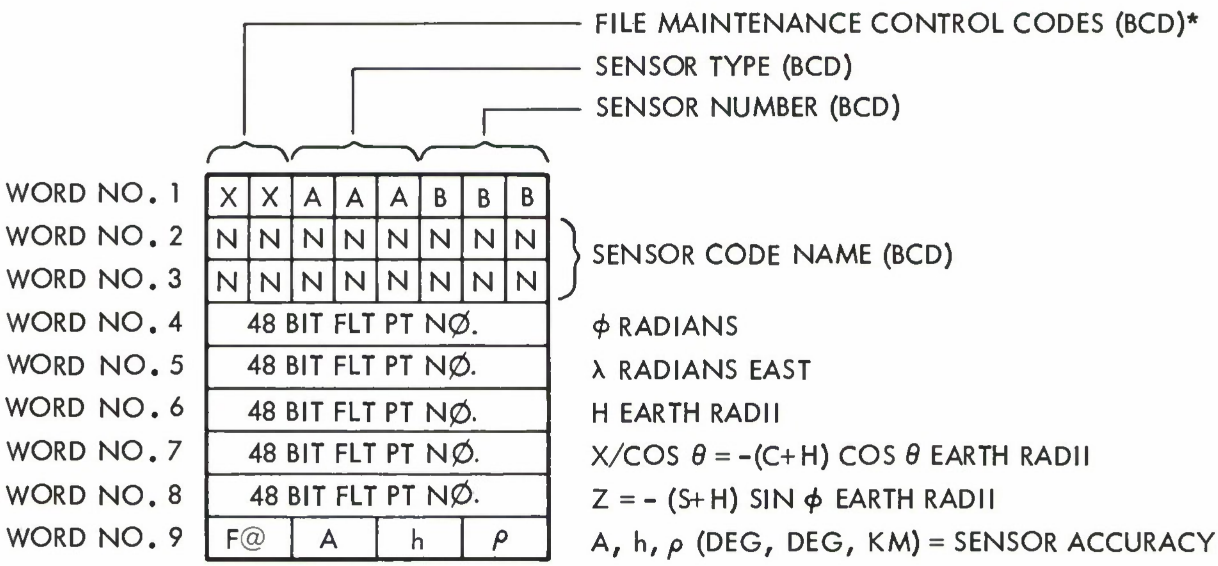


Figure 3-2. SEAI Tape Format

3.7.2.1 Sensor Format



*XX = 11 (BCD) FOR SENSOR FILE

WORD NO. 9

BIT LAYOUT FOR WORD NO. 9

BITS 0-5	F	SIGN BIT ON = CLASSIFIED
		BIT 5 ON = NOT REPORTING
BITS 6-11	@	LA COORDINATES INDICATOR (BCD) C TYPE (ONLY THE LEAST SIGNIFICANT 04 BITS ARE USED)
BITS 12-23	A	
BITS 24-35	h	
BITS 36-47	ρ	

14 RECORDS PER BLOCK = 125 WORDS (2 WORDS NOT USED)

A, h, ρ, F ARE ALL ZERO AT THIS TIME

Figure 3-3. Sensor Format

3.7.2.2 Element Record

WORD NO. 1	X X | N N | O B B B	X X: FILE MAINTENANCE CONTROL*; N N: FIRST TWO CHARACTERS OF SATELLITE NAME; O B B B: SATELLITE NUMBER
WORD NO. 2	N N N N N N N N	LAST 8 CHARACTERS OF SATELLITE NAME
WORD NO. 3	48 BIT FLT PT NØ.	a_{xn}
WORD NO. 4	48 BIT FLT PT NØ.	a_{yn}
WORD NO. 5	48 BIT FLT PT NØ.	hx
WORD NO. 6	48 BIT FLT PT NØ.	hy
WORD NO. 7	48 BIT FLT PT NØ.	hz
WORD NO. 8	48 BIT FLT PT NØ.	L_o RADIUS
WORD NO. 9	48 BIT FLT PT NØ.	T_o DAYS SINCE 1950
WORD NO. 10	48 BIT FLT PT NØ.	C DAYS/REV^2
WORD NO. 11	48 BIT FLT PT NØ.	RMS KILOMETERS
WORD NO. 12	48 BIT FLT PT NØ.	NO. OF OBSERVATIONS IN RMS
WORD NO. 13	48 BIT FLT PT NØ.	
WORD NO. 14	INITIAL REV | EPOCH REV | FINAL REV	2-24 BIT BINARY INTEGERS
WORD NO. 15	C_p | STANDARD BRIGHTNESS	STORED AS INTEGERS X 10+
WORD NO. 16	NOT USED | I | T_{oy} | Y Y Y	I: ISTOP, (BCD); T_{oy}: YEAR OF T_o - (1 BCD CHARACTER); Y Y Y: ELEMENT SET NUMBER - BINARY INTEGERS

*2 BCD CHARACTERS - ELEMENT FILE = 13

WORD NO. 13 - EXPIRATION DATE OF BULLETIN - DAYS AND FRACTIONS OF A DAY

WORD NO. 14 - INITIAL REV- BITS 0-15
EPOCH REV- BITS 16-31
FINAL REV - BITS 32-47

WORD NO. 16 - BIT O OF WORD NO. 16 IS THE SIGN OF THE STANDARD BRIGHTNESS

Figure 3-4. Element Record

3.7.3 SRADU Tape Format

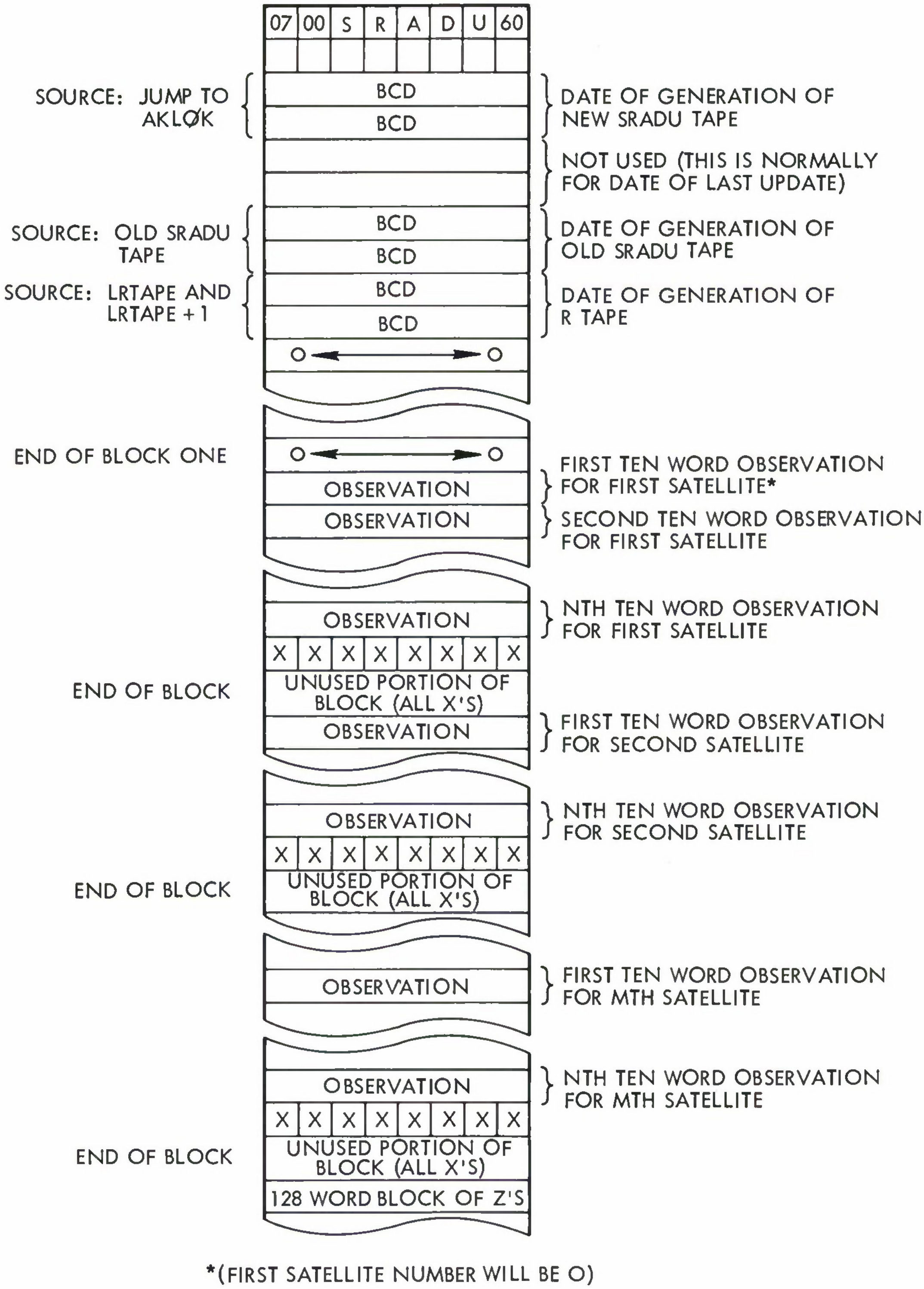
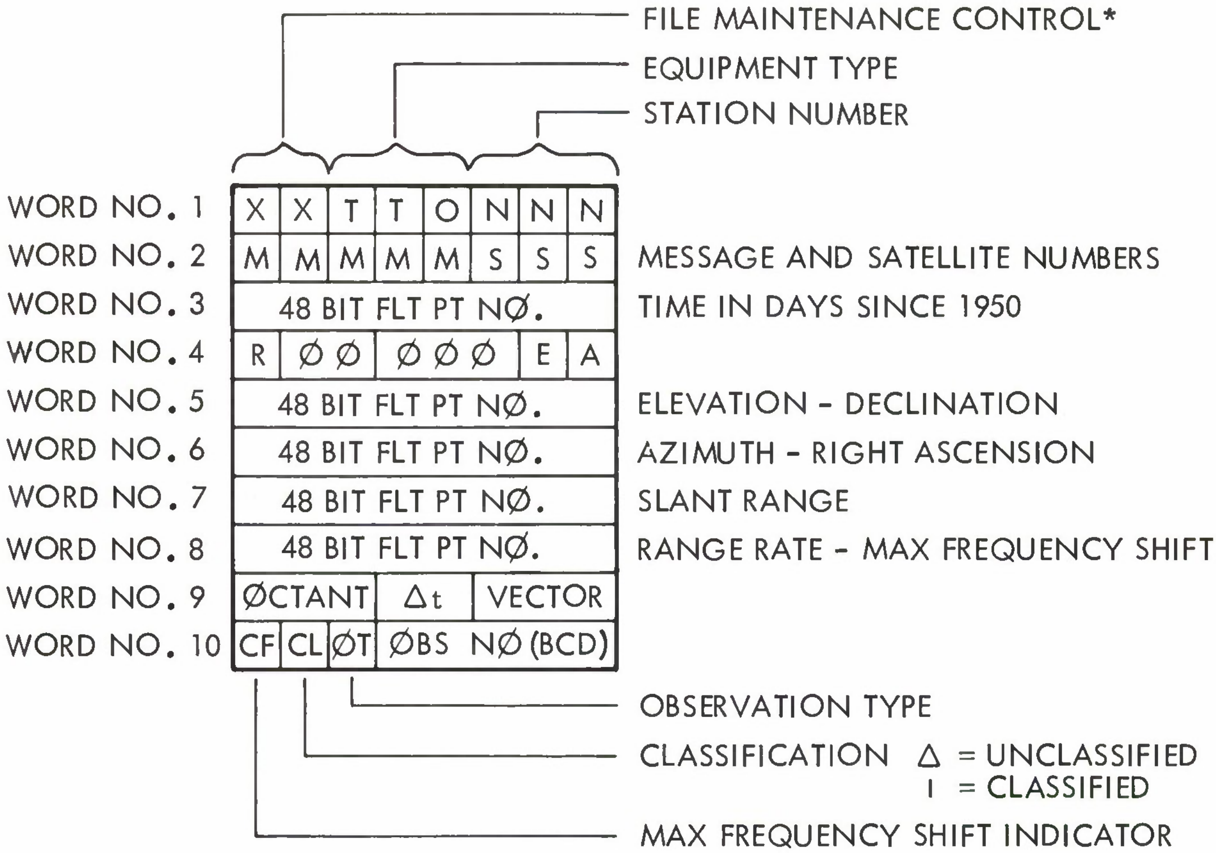


Figure 3-5. SRADU Tape Format

3.7.3.1 <u>Observation Record</u>



*2 BCD CHARACTERS - OBSERVATION FILE = 17

WORD NO. 4

R = ASSOCIATION INDICATOR
(1 BCD CHARACTER -1-9)

Ø = O NOT USED

E = EQUINOX

A = ACCURACY

WORD NO. 9

OCTANT (DEGREES)	BITS 0-15
Δt X 100 (MIN)	BITS 16-31
VECTOR MAGNITUDE	BITS 32-47

Figure 3-6. Observation Record

3.7.4 Scratch Tape (Log No. 7)

The first block on tape 7 is the identification block. The first word on the block is the tape identification. The second word is the vehicle number. The remainder of the block contains blanks. The identification block is shown in Figure 3-7.

The next 60 blocks on tape 7 contain CØMMØN storage. All the words in all the blocks are 48 bit floating point numbers.

After CØMMØN is written, a sentinel block is written, consisting of words of z's.

Blocks of observation information follow the sentinel block. A block of observation information is written according to the format on diagram B. Tape 7 is filled with observations from the sentinel block until the end of the tape.

3.7.4.1 Identification Block of Tape Log 7

WORD NO. 1	7	0		T	A	P	E	7	TAPE IDENTIFICATION
WORD NO. 2	O	X	X	X	Δ	Δ	Δ	Δ	VEHICLE NO.

X X X = VEHICLE NO.

Figure 3-7. Tape Log 7 Identification Block

3.7.4.2 Tape 7 Observation Format

G_s - BINARY INTEGER AT B23

SENSOR NUMBER (BCD)

WORD NO. 1	G_s				O	N	N	N	
WORD NO. 2	CF	CL	ØT	ET		S	S	S	
WORD NO. 3	48 BIT FL PT NO.								TIME - DAY AND FRACTIONS OF DAYS FROM 1 JANUARY 1950
WORD NO. 4	R	Ø	Ø	Ø	Ø	Ø	E	A	
WORD NO. 5	48 BIT FL PT NØ								ELEVATION - DECLINATION
WORD NO. 6	48 BIT FL PT NØ								AZIMUTH - HOUR ANGLE
WORD NO. 7	48 BIT FL PT NØ								SLANT RANGE
WORD NO. 8	48 BIT FL PT NØ								RANGE RATE
WORD NO. 9	σ_R				σ_A				OBSERVATION WEIGHTS ASSIGNED AT OBSERVATION PROCESSING TIME*
WORD NO.10	σ_E				$\sigma_{\dot{R}}$				

*THESE WEIGHTS ARE STORED AS BINARY INTEGERS, TWO PER WORD (ONE AT B23 AND THE OTHER AT B47). THE TRUE WEIGHTS ARE THESE INTEGERS CONVERTED TO FLOATING POINT NUMBERS AND DIVIDED BY 10^4. FOR OPTICAL DATA THE FIRST WORD CONTAINS WEIGHTS FOR FIELD REDUCED RA AND DEC AND THE SECOND WORD CONTAINS WEIGHTS FOR PRECISION REDUCED RA AND DEC

WORD NO. 2

CF = MAX FREQUENCY SHIFT INDICATOR

CL = CLASSIFICATION Δ = UNCLASSIFIED 1 = CLASSIFIED

ØT = OBSERVATION TYPE

- 0 = RANGE RATE ONLY
- 1 = AZIMUTH AND ELEVATION
- 2 = AZIMUTH, ELEVATION, AND RANGE
- 3 = AZIMUTH, ELEVATION, RANGE AND RANGE RATE
- 5 = RIGHT ASCENSION AND DECLINATION

ET = EQUIPMENT TYPE

WORD NO. 4

R = ASSOCIATION INDICATOR

E = EQUINOX

A = ACCURACY

Figure 3-8. Tape 7 Observation Format

3.7.5 Binary Ephemeris Tape

WORD NO. 1	O	X	X	X	Δ	Δ	Δ	Δ	VEHICLE NO.
WORD NO. 2									VEHICLE NAME (BCD)
WORD NO. 3									

(a)

T = TIME IN MINUTES FROM 0^h DAY OF EPOCH

X = (E.R.)

Y = (E.R.)

Z = (E.R.)

$\dot{X}$ = (E.R./KEMIN)

$\dot{Y}$ = (E.R./KEMIN)

$\dot{Z}$ = (E.R. / KEMIN)

(b)

THE TRAJECTORY TAPE IS WRITTEN, OPTIONALLY, ON LOGICAL TAPE UNIT 10. THE FIRST BLOCK ON TAPE CONTAINS ALL BLANKS EXCEPT FOR THE FIRST 3 WORDS. THE FORMAT OF THE FIRST 3 WORDS IS SHOWN IN (a). EIGHTEEN SETS, OF THE FORM IN (b), MAKE UP A BLOCK OF INFORMATION ON THE TRAJECTORY TAPE. A SET CONSISTS OF SEVEN FLOATING POINT NUMBERS

A SENTINEL BLOCK CONTAINING ALL Z'S FOLLOWS THE FINAL TRAJECTORY BLOCK WHICH IS WRITTEN.

Figure 3-9. Binary Ephemeris Tape

4. ESPOD SUBROUTINE DESCRIPTION

This section identifies and describes each subroutine used in the ESPOD program. The segments of the program which use these subroutines are discussed in Section 3.3 and glossaries giving abbreviated descriptions are provided for each of the segments: ESPØD, ESPØDDC and ESPØDEPH.

Each subroutine is described in the following terms:

a) Identification—title, segment, called by subroutine

b) Function

c) Usage—calling sequence, input, output, error/action messages on the line

d) Subroutines used—library, program

e) Equations

The subroutines are presented in alphabetical order by title. A complete abbreviated alphabetical listing of titles with page number is provided for ready reference.

4.1 ALPHABETICAL LISTING OF TITLES

4. 2 SUBROUTINE DESCRIPTIONS

SUBROUTINE IDENTIFICATION

A. Title

ADJUST

B. Segment

ESPØD

C. Called by subroutine

PRECES

FUNCTION

The function is to update right ascension, declination type observations to true equinox of midnight of the day of epoch.

USAGE

A. Calling sequence

Call ADJUST(D, E, C)

B. Input

1. CØMMØN

CDEG	Degrees/radian
CSEPS	Cos ϵ
C2PI	2π
DDAY	Epoch day
DLEPS	$\Delta\epsilon$
DLPSI	$\Delta\psi$
DMNTH	Epoch month
DYEAR	Epoch year
SNEPS	Sin ϵ

2. Calling sequence

D	Observed value of right ascension
E	Observed value of declination
C	i (the reference year for the observation)

C. Output

1. CØMMØN

——

2. Calling sequence

D	Value of right ascension which has been precessed
E	Value of declination which has been precessed

D. Error/action messages

——

SUBROUTINES USED

A. Library

ABSF
CØSF
SINF
TANF

B. Program

ATNQF Arc tangent routine

EQUATIONS

Update α, δ observations to equinox of 0000Z day of epoch

$$t_o = 1900 + Y + \frac{M}{12} + \frac{D}{365.25}$$

$$T_o = \left[\frac{i_{years} - 1900}{100}\right]$$

$$T = \frac{\left[t_o - i\right]}{100}$$

$$\xi_o = (2304.25'' + 1.396''T_o)\left|T\right| + 0.302''\left|T\right|^2 + 0.018''\left|T\right|^3$$

$$z = \xi_o + 0.7911''\left|T\right|^2$$

$$\theta = (2004.682'' - 0.853''T_o)\left|T\right| - 0.426''\left|T\right|^2 - 0.042''\left|T\right|^3$$

if $T \leq 0$, continue

if $T > 0$, go to (I)

$$\xi_o = -z$$

$$z = -\xi_o$$

$$\theta = -\theta$$

$$(I) = \cos\delta_o \sin(\alpha_o + \xi_o)$$

$$(II) = \cos\theta \cos\delta_o \cos(\alpha_o + \xi_o) - \sin\theta \sin\delta_o$$

$$(\mathrm{III}) = \cos\theta \sin\delta_o + \sin\theta \cos\delta_o \cos(\alpha_o + \xi_o)$$

$$\Delta\alpha = (\cos\epsilon + \sin\epsilon \sin\alpha \tan\delta)\Delta\psi - \cos\alpha \tan\delta\Delta\epsilon$$

$$\Delta\delta = \sin\epsilon \cos\alpha\Delta\psi + \sin\alpha\Delta\epsilon$$

$$\alpha = \tan^{-1}\left[\frac{(\mathrm{I})}{(\mathrm{II})}\right] + Z + \Delta\alpha$$

$$\delta = \tan^{-1}\left[\frac{\frac{(\mathrm{III})}{(\mathrm{II})}}{\cos(\alpha - z)}\right] + \Delta\delta$$

SUBROUTINE IDENTIFICATION

A. Title

ALSØRT

B. Segment

ESPØD

C. Called by subroutine

LØDSEN

FUNCTION

This routine will sort alphanumerically the list of desired sensor numbers. This list is generated when the observations are being processed.

USAGE

A. Calling sequence

Call ALSØRT

B. Input

1. CØMMØN

DBUFS	Auxiliary buffer storage
TEMP	Temporary storage

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

APF10

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

CØESA (ESPØDDC, ESPØDEPH)
PAETZØLD (ESPØDDC, ESPØDEPH)

FUNCTION

The function is to compute values of A_p and F_{10} as a function of time. Values of A_p and F_{10} are used by dynamic atmosphere routines in their computations of density. A table consisting of sets of t (days), A_p, F_{10} are input with the preliminary data. Linear interpolation is used where possible, and where it is not possible, the last values of A_p and F_{10} in the table are used. New values are computed only if the time has changed by more than a quarter of a day from the time of the last computation. The input table is first checked for over-the-year discontinuities in time. If discontinuities appear, the times in the table are adjusted appropriately.

USAGE

A. Calling sequence

Call APF10

B. Input

1. CØMMØN

CAPF10	Array containing sets of t (days), A_p, F_{10} Maximum of 30 sets is permissible
TLIST(2)	Time (min from 0^h day of epoch) for which to compute values of A_p and F_{10}
DFL	Flag to indicate first time in

2. Calling sequence

—

C. Output

1. CØMMØN

CAP	A_p for time TLIST(2)
CF10	F_{10} for time TLIST(2)

2. Calling sequence

——

D. Error/action messages

SUBROUTINES USED

A. Library

——

B. Program

——

SUBROUTINE IDENTIFICATION

A. Title

APPLY

B. Segment

ESPØDDC

C. Called by subroutine

FIT

FUNCTION

Function is to apply DC solution vector and print iteration summary.

USAGE

A. Calling sequence

Call APPLY (IFIT)

B. Input

1. CØMMØN

CDAD2M	$C_DA/2m$
CK	Drag variation
IVSTR	Fixed point variable storage
NBDNS	Starting location of bounds vector in variable storage
NDPAR1	Starting location of solution vector in variable storage
NDPR	Total number of Category 1 variables to solve for
NICPR	Total number of spherical coordinates to solve for
NIDP	Starting location in fixed point variable storage of an array which defines CAT1 variables in solution vector
NITCT	Iteration counter
NPAR	Starting location of parameter list in variable storage

NPBIS	Starting location of current estimates of Category 2 variables
NPR	Total number of parameters to solve for
NPRCD	Identifies table for definition of Category 2 variables to be solved for
NSCALE	Starting location of the list of conversion factors
NSSTB	Starting location where station mean and RMS information are stored
NSTAT	Starting location of the master sensor table
NR	Starting location of where the $(A^TA)^{-1}$ is stored
NRTMP	Starting location of temporary storage for special handling of R matrix
TICRT	Nominal Cartesian coordinates
TIPØL	Nominal spherical coordinates
TEMP	Temporary storage
TNØMP	Initial spherical coordinates
TNØMX	Initial Cartesian coordinates
TSUS	Current total SØS
TSUSB	Best SØS so far
TSUSP	Predicted SØS for next iteration
TZ	Indicates if solution was affected by bounds
VSTR	Variable storage
CDEG	Degree/radian
CKMER	Km/Earth radii
IØUT	Output tape number

2. Calling sequence

IFIT	1 apply solution using nominal bounds 2 apply solution using bounds over two 3 apply solution using bounds over four 4 apply solution using bounds over eight

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/ action messages

—

SUBROUTINES USED

A. Library

SQRTF

B. Program

CØRMAT	Computes correlation (σ and ρ) matrix
HUMAH	Converts vector or matrix from machine units to human units
MATPT	Prints an N x N lower triangular matrix
MØVMAT	Moves a triangular matrix from A to B storage
PTØC	Converts polar to Cartesian

SUBROUTINE IDENTIFICATION

A. Title

ASIN

B. Segment

ESPØD
ESPØDDC
ESPØDEPH

C. Called by subroutines

—

FUNCTION

The function is to compute the arc sine in radians between $-\pi/2$ and $\pi/2$.

USAGE

A. Calling sequence

ASIN (A)

B. Input

1. CØMMØN

—

2. Calling sequence

A	Argument between -1.0 and +1.0

C. Output

1. CØMMØN

—

2. Calling sequence

ARCSIN	Radians (principal value)

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRTF

B. Program

ATNQF	Arc tangent

METHOD

—

EQUATIONS

—

SUBROUTINE IDENTIFICATION

A. Title

ASSIGN

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

The function is to establish NPR, NDPR, NICPR, NIDENT and do the storage assignment for the arrays to be located in VSTR and IVSTR.

USAGE

A. Calling sequence

Call ASSIGN

B. Input

1. CØMMØN

CLDSTR	Cold start, non-cold start flag
DATA	Input storage
DCFLG	DC package control flags
FGCAT1	Flag to indicate category 1 card read
FGCAT2	Flag to indicate category 2 card read
FGDELE	Flag to indicate delete cards read
NARØW	Starting location where one row of the augmented matrix (A, B) is stored
NATA	Starting location of where the triangular A^TA is stored
NBDNS	Starting location for the bounds used by LEGS
NDPAR1 NDPAR2 NDPAR3 NDPAR4	Starting locations where the four sets of solution vectors will be stored

NDPR	Number of all differential and initial parameters to solve for (Category 1)
NICPR	Number of initial conditions parameters to solve for
NIDENT	Number of entries in the NIDLED list
NIDLED	Starting location of where the observation deletion table begins
NIDP	Identifier for table indicating Category1 type variables to be solved for
NPAR	Identifies the starting location for the parameter list
NPBIS	Identifies table for current estimates of Category 2 variables
NPR	Number of all parameters to solve for
NPRCD	Identifies table for definition of Category 2 variables to be solved for
NR	Starting location of where the inverse $A^T A$ (in triangular form) is stored
NRTMP	Identifies the starting location of temporary storage for special handling of the R matrix
NSCALE	Starting location of the list of conversion factors which convert all solution vectors and associated matrices from machine to output units and vice versa
NSMAT	Identifies starting location of *a priori* S matrix
NSSTB	Identifies starting location where station information concerning computed sigmas and means of residuals are stored
NSTAT	Starting location of the master sensor table
NUBS	Identifies the starting location of the observation table
VSTR	Floating point variable storage

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

MØVE — Moves blocks of storage n cells either forward or backward in core

EQUATIONS

NICPR = Number of orbital elements to solve for

NDPR = CAT1 variables

NPR = CAT1 + CAT2

NIDP = 1

NPRCD = NDPR + NIDP

NPBIS = NPR - NDPR + NPRCD

NARØW = NPR - NDPR + NPBIS

NBDNS = NPR - NARØW + 1

NPAR = NPR + NBDNS

NDPAR1 = 2*NPR + NPAR

NDPAR2 = NPR + NDRAR1

NDPAR3 = NPR + NDPAR2

NDPAR4 = NPR + NDPAR3

NSCALE = NPR + NDPAR4

$$NIDLED = NPR + NSCALE$$

$$NATA = NIDENT + 2 + NIDLED$$

$$NR = \left[(NPR + 1) * (NPR + 2)\right]/2 + NATA$$

$$NRTMP = \left[(NPR + 2) * (NPR + 3)\right]/2 - 1 + NR$$

$$III = \left[NPR * (NPR + 1)\right]/2$$

$$NSMAT = III + NRTMP + 1$$

If $\left[DCFLG(2)\right] = 0$, set $NSTAT = NSMAT + 1$

If $\left[DCFLG(2)\right] \neq 0$, set $NSTAT = III + NSMAT + 1$

SUBROUTINE IDENTIFICATION

A. Title

ATNQF

B. Segment

ESPØD

ESPØDDC

ESPØDEPH

C. Called by subroutine

—

FUNCTION

The function is to obtain, using ATANF, arc tan X, where X = A/B, given A and B. The range of ATNQF is $-\pi$ and π.

USAGE

A. Calling sequence

ATNQF (A, B)

B. Input

1. CØMMØN

   CPI π

2. Calling sequence

   A/B in radians

C. Output

1. CØMMØN

   —

2. Calling sequence

   $-\pi \leq X \leq + \pi$ in radians

D. Error/action messages

—

SUBROUTINES USED

A. Library

ATANF

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

ATMØS

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

DRAG (ESPØDDC, ESPØDEPH)

FUNCTION

The function is to drive for density calculation.

USAGE

A. Calling sequence

Call ATMØS

B. Input

1. CØMMØN

CDRAGM Input flag to indicate which model atmosphere is to be used

CDRAGM = 1) ARDC 1959

2) Paetzold dynamic

3) CØESA static

4) CØESA dynamic

If CDRAGM = 0, the CØESA static atmosphere is used

C. Output

1. CØMMØN

—

2. Calling sequence

—

SUBROUTINES USED

A. Library

—

B. Program

ATM59	Static atmosphere
DYNAT	Dynamic atmosphere
CØESA	Dynamic and static atmosphere

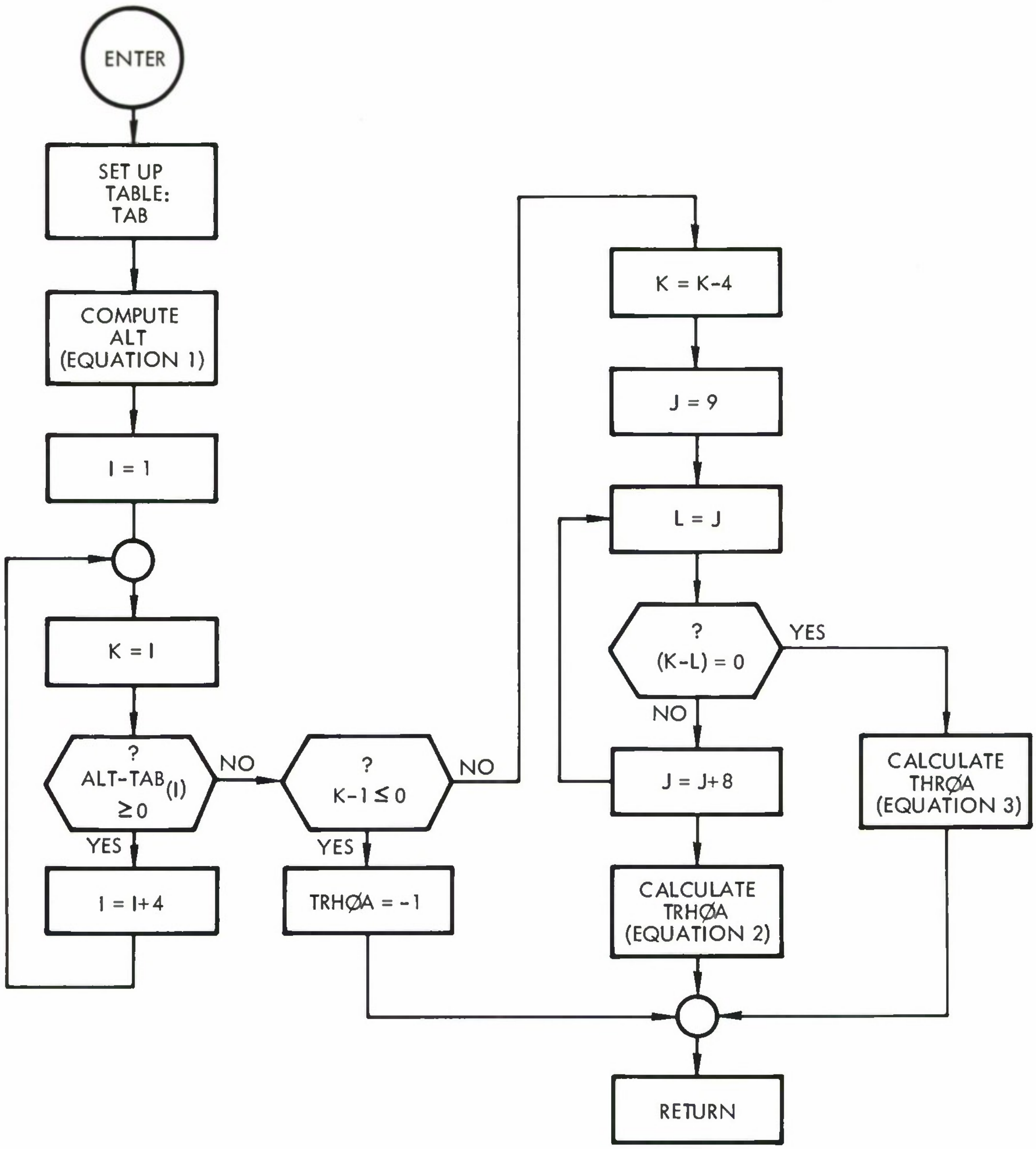


Figure 4-1. ATM59 Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

ATM59

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutine

ATMØS (ESPØDDC)

DYNAT (ESPØDEPH)

FUNCTION

The function is to interpolate from the atmosphere tables the density of the atmosphere at given altitudes, using the standard ARDC 1959 model.

USAGE

A. Calling Sequence

Call ATM59

B. Input

1. CØMMØN

TALT Altitude (meters)

2. Calling sequence

—

C. Output

1. CØMMØN

TRHØA Density (kg/m^3)

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

$$H = \frac{g_o}{G}\left[\frac{rz}{r+Z}\right] \qquad (1)$$

$$\rho = \rho_b \left[\frac{(T_M)_b}{(T_M)_b + L_M (H - H_b)}\right]^{1+(GM_o/R^* L_M)} \qquad \text{for } L_M \neq 0 \qquad (2)$$

$$\rho = \rho_b \exp\left[\frac{-GM_o(H - H_b)}{R^*(T_M)_b}\right] \qquad \text{for } L_M = 0 \qquad (3)$$

where b refers to the value of the quantity at the base of the constant gradient layer.

Note:

Equation (1)

H = geopotential altitude

g_o = acceleration of gravity

G = conversion constant

$= \dfrac{9.80665\ M^2}{\sec^2\ M^1}$ where M^1 is meters of geopotential

r = effective Earth radius at latitude $45^\circ 32' 33''$

Z = geometric altitude

Equations (2) and (3)

ρ = Density obtained from calculation

ρ_b = density at the base of a constant gradient layer where these base values were obtained.†

$(T_M)_b$ = molecular-scale temperature at the base of a constant gradient layer.†

†R.A. Minzner, K.S. Champion, and H. L Pond, The ARDC Model Atmosphere, 1959 Air Force Surveys in Geophysics No. 115 (AFCRC-TR-59-267) Air Force Cambridge Res. Center, August 1959.

 ATM59

L_M = molecular scale temperature gradient

$$= \frac{T_M - (T_M)_b}{H - H_b}$$

M_o = sea level value of molecular weight

R^* = universal gas constant

SUBROUTINE IDENTIFICATION

A. Title

BCDØBS

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

The function is to read in one observation card and to pack the information into a format identical to an observation format read in on the SRADU tapes.

USAGE

A. Calling sequence

Call BCDØBS (SEØF)

B. Input

1. CØMMØN

—

2. Calling sequence

SEØF Sentinel block detection flag

C. Output

1. CØMMØN

TEMP(30)	Satellite number	(A)	*
(31)	Equipment type	(A)	
(32)	Station number	(A)	
(33)	Year		
(34)	Month		
(35)	Day		
(36)	Hour		
(37)	Minutes		
(38)	Seconds		
(39)	E or δ		
(40)	A or α		

(A) = Alphanumeric

* Indicates packed information

(41)	R		
(42)	$\dot{R}$		
(43)	Code for $\dot{R}$	(A)	
(44)	At observation time	(A)	*
(45)	Maximum } Brightness		*
(46)	Minimum } Brightness		*
(47)	Time interval		
(48)	Date or line number	(A)	
(49)	Message number	(A)	*
(50)	Equinox	(A)	*
(51)	Year	(A)	
(52)	Observation number	(A)	
(53)	Card type	(A)	

2. Calling sequence

—

D. Error/action messages

1. Off-line comment:

"THE FOLLOWING CARD(S) COULD NOT BE CONVERTED ____ ERR LOCATION."

2. Action

NONE

SUBROUTINES USED

A. Library

GLØP

B. Program

IDSUB Strips blanks from I.D.

XSRCH Card image scan and convert

(A) = Alphanumeric

* Indicates packed information

SUBROUTINE IDENTIFICATION

A. Title

BØDY

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

DAUX (ESPØDDC, ESPØDEPH)

FUNCTION

The function is to compute the perturbative acceleration of a spacecraft due to other bodies in the solar system and to account for these effects in the variational equations.

USAGE

A. Calling sequence

Call BØDY

B. Input

1. CØMMØN

BFLAGS	Flags to indicate which bodies are to be considered
TLIST	Current integration list
DBASE	Days from 1950.0 to midnight day of epoch
CMU	GM of Earth (e. r.^{3}/min^2)
CGMR	Ratio of Earth, moon, sun, Venus, Mars, Saturn and Jupiter GM to that of the Earth
FLVE	Flag to skip computation of variational equations
NDPR	Total number of Category 1 variables to solve for

2. Calling sequence

—

C. Output

1. CØMMØN

TBPERT	The total acceleration of the vehicle due to all the desired bodies
PMAT	Matrix of the position dependent effects in the variational equations (the body effects are added to this matrix)

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

INTPL

MAGN

ØUTER

EQUATIONS

The position of the ith body with respect to the Earth, x_i, y_i, z_i, is obtained from the ephemeris tape.

$$R_i = \left(x_i^2 + y_i^2 + z_i^2 \right)^{1/2}$$

$$x_{vi} = x_v - x_i$$

$$y_{vi} = y_v - y_i$$

$$z_{vi} = z_v - z_i$$

where x_v, y_v, z_v is the position of the vehicle with respect to the earth.

$$R_{vi} = \left(x_{vi}^2 + y_{vi}^2 + z_{vi}^2 \right)^{1/2}$$

$$\ddot{x}_{bodies} = - \sum_{i=1}^{u} \mu_i \left[\frac{(x_v - x_i)}{R_{vi}^3} + \frac{x_i}{R_i^3} \right]$$

$$\ddot{y}_{bodies} = -\sum_{i=1}^{u} \mu_i \left[\frac{(y_v - y_i)}{R_{vi}^3} + \frac{y_i}{R_i^3}\right]$$

$$\ddot{z}_{bodies} = -\sum_{i=1}^{u} \mu_i \left[\frac{(z_v - z_i)}{R_{vi}^3} + \frac{z_i}{R_i^3}\right]$$

$$\text{PMAT} = \text{PMAT} + \begin{bmatrix} \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}^2}{R_{vi}^5} - \frac{1}{R_{vi}^3}\right) & \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, y_{vi}}{R_{vi}^5}\right) & \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, z_{vi}}{R_{vi}^5}\right) \\ \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, y_{vi}}{R_{vi}^5}\right) & \sum_{i=1}^{u} \mu_i \left(\frac{3y_{vi}^2}{R_{vi}^5} - \frac{1}{R_{vi}^3}\right) & \sum_{i=1}^{u} \mu_i \left(\frac{3y_{vi}\, z_{vi}}{R_{vi}^5}\right) \\ \sum_{i=1}^{u} \mu_i \left(\frac{3x_{vi}\, z_{vi}}{R_{vi}^5}\right) & \sum_{i=1}^{u} \mu_i \left(\frac{3y_{vi}\, z_{vi}}{R_{vi}^5}\right) & \sum_{i=1}^{u} \mu_i \left(\frac{3z_{vi}^2}{R_{vi}^5} - \frac{1}{R_{vi}^3}\right) \end{bmatrix}$$

SUBROUTINE IDENTIFICATION

A. Title

BØUNDS

B. Segment

ESPØDDC

C. Called by subroutines

FIT

FUNCTION

The function is to scale bounds with a given scale factor.

USAGE

A. Calling sequence

Call BØUNDS (SCALE)

B. Input

1. CØMMØN

NBDNS	Starting location for the bounds in variable storage
NPR	Number of all parameters to solve for

2. Calling sequence

SCALE	Scale factor for bounds

C. Output

1. CØMMØN

VSTR (NBONS) Contains bounds which are scaled

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

$B_i = K\ B_i \quad K = \text{scale}$

SUBROUTINE IDENTIFICATION

A. Title

CALCSG

B. Segment

ESPØD

C. Called by subroutine

SWTSN

FUNCTION

Subroutine CALCSG calculates the CSIG table entries and stores them. The sigmas are computed as a function of credance as given by the particular observation.

USAGE

A. Calling sequence

Call CALCSG (A, I, C)

B. Input

1. CØMMØN

—

2. Calling sequence

C Credance

C. Output

1. CØMMØN

—

2. Calling sequence

A(I) = σ_R, σ_A

A(I + 1) = σ_E, $\sigma_{\dot{R}}$

Sigmas are packed as follows:

Word 1	σ_R	σ_A
Word 2	σ_E	$\sigma_{\dot{R}}$

The sigmas are binary integers scaled by 10^4 with binary points at 23 and 47.

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

$$\sigma_{A,E} = \frac{0.26}{1 + 0.51\,c + 0.075\,c^2} \quad \text{deg}$$

$$\sigma_R = \frac{43}{1 + 0.81\,c + 0.582\,c^2} \quad \text{km}$$

$$\sigma_{\dot{R}} = \frac{0.07}{1 + 4.848\,c - 0.115\,c^2} \quad \text{km/sec}$$

SUBROUTINE IDENTIFICATION

A. Title

CLTIME

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

Function is to take the time in days and fractions of days from 1950.0 and compute the calendar date.

USAGE

A. Calling sequence

Call CLTIME(TG)

B. Input

1. CØMMØN

CDAYMN Number of days in the month

2. Calling sequence

TG Time in days and fraction of days from 1950

C. Output

1. CØMMØN

TEMP(3) Year
TEMP(4) Month
TEMP(5) Day
TEMP(6) Hour
TEMP(7) Minutes
TEMP(8) Seconds

2. Calling sequence

—

D. Error/action messages

Action: Subroutine error is called if TG (time in days and fractions of days from 1950.0) is negative or less than 1950.

SUBROUTINES USED

A. Library

B. Program

ERRØR Error return

SUBROUTINE IDENTIFICATION

A. Title

CØESA

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

DRAG (ESPØDDC, ESPØDEPH)

FUNCTION

Function is to compute the density at a given altitude using U. S. Standard Atmosphere, 1962, as the model. This routine computes density from either a static model or a dynamic model if appropriate parameters A_p and F_{10} are input.

USAGE

A. Calling sequence

Call CØESA

B. Input

1. CØMMØN

TALT	Altitude in meters
CAP	A_p for this time
CF10	F_{10} for this time
TALFAG	Sidereal time at 0^h day of epoch
TLIST(2)	Time in minutes from 0^h day of epoch
TLIST(4)	x
TLIST(5)	y
TLIST(6)	z
TR	Radius at this time
CDEG	Degrees/radian
CPI	π
CWE	Rotation rate of the Earth (rad/min)
DFL	Flag for first time in

2. Calling sequence

—

C. Output

1. CØMMØN

THRØA density (kg/m^3)

2. Calling sequence

D. Error/ action messages

—

SUBROUTINES USED

A. Library

EXPF

LØGF

SINF

CØSF

B. Program

APF10 Computes A_p, F_{10} for t

EQUATIONS

Reference 1.*

This atmosphere was divided into two regions:

Region 1 extends from -5 km to 90 km

Region 2 extends from 90 km to 700 km

Within these regions the atmosphere was further divided into layers of constant gradient of molecular scale temperature with altitude. See Table 1.4 (e) for the values of T_m, L_m and T at the base of each layer.

For Region 1:

Density was determined from the equations:

$$1. \quad \rho = \rho_b \left[\frac{\left(T_m\right)_b}{\left(T_m\right)_b + L_m \left(H - H_b\right)} \right]^{1 + \frac{GM_o}{R^* L_m}} \qquad \text{for } L_m \neq 0$$

where

GM_o = gravitational parameter

R^* = gas constant

= 8.31432 joules/°K - mole

*Reference 1. United States Committee on Extension to the Standard Atmosphere (CØESA), U. S. Standard Atmosphere, 1962, Washington, D. C., 1962.

T_m = molecular scale temperature

L_m = gradient $^{o}K/km$

H = geopotential altitude

and b refers to the value at the base of the particular layer.

$$2. \quad \rho = \rho_b \exp\left[-\frac{GM_o\,(H - H_b)}{R^*\left(T_m\right)_b}\right] \qquad \text{for } L_m = 0$$

The pressure equivalent of Equations 1 and 2 are in Reference 1., Equations I. 2. 10-3 and I. 2. 10-4.

In order to avoid performing the integration for geopotential altitude

$$H = \frac{1}{g_o}\,\Phi = \int_o^z \frac{g}{g_o}\,dz \quad \text{(Reference 1, Equation I. 2. 5-1)}$$

a polynomial curve was determined.

$$3. \quad H(z) = 0.99999352\ z - 0.15700906 \times 10^{-6}\ z^2 + 0.21277556 \times 10^{-13}\ z^3$$

where z is the geometric altitude.

For Region 2:

Beyond 90 k_m T_m is a linear function of geometric altitude.

$$T_m = \left(T_m\right)_b + L_n\,(z - z_b) \quad \text{(Reference 1, Equation I.2.6-3)}$$

The pressure is given by Reference 1, Equation I.2.10-5.

$$4. \quad \log_e P = \log_e P_b - \frac{1}{L_m}\frac{M_o}{R^*}\int_{z_o}^{z} \frac{g\,dz}{z - z_b + \dfrac{\left(T_m\right)_b}{L_m}}$$

in order to evaluate this integral a polynomial

5. $g(z) = A + B z + C z^2 + D z^3$

was employed, where

$$A = 9.806853$$
$$B = -0.3087135 \times 10^{-5}$$
$$C = 0.7193415 \text{ c } 10^{-12}$$
$$D = -0.1236578 \times 10^{-18}$$

For both Regions 1 and 2 the program picks up the proper base values from the table, TAB and solves Equations 1 through 4. Excluding the equations the program logic is concerned with picking the proper values from the table.

Between the temperature range of 800-2100 $^{\circ}$K and the altitude range of 200-700 km and when values of F_{10} and a_p are available the atmosphere solves equations to obtain a correction of $\log_{10} \rho$ to make the atmosphere model semi-dynamic.

at start

$$\alpha_g = \alpha_{go} \tag{1}$$

at start of each day

1) Find the subsolar point $\alpha_s = \hat{\alpha}_g - 180^{\circ}$ (2)

$$\sin \delta_s = \sin (23.5) \sin \alpha_s \tag{3}$$

$$\cos \delta_s = \sqrt{1 - \sin^2 \delta_s} \tag{4}$$

2) Find subbulge point

$$\alpha_B = \alpha_s + 30^{\circ} \quad (\delta_B = \delta_s) \tag{5}$$

Let

$$y_1 = \cos \delta_B \cos \alpha_B \tag{6}$$

$$y_2 = \cos \delta_B \sin \alpha_B \tag{7}$$

$$y_3 = \sin \delta_B \tag{8}$$

then

$$\bar{y} = y_1 + y_2 + y_3$$

is a unit vector directed at the subbulge point.

3) Find T_N (Reference 1, Equation II.2.3-5)

$$T_N = 1025 + 4.5\,(F_{10} - 170) + 0.5 \cos 2\left(\tilde{a}_g - \frac{195.298}{57.29577}\right) + 1.5\, a_p \qquad (9)$$

$$T = T_N\left[1 + 0.4\left(\frac{1 + \cos \psi}{2}\right)^{2.5}\right]$$

which is the half angle equivalent of Equation II. 2.3-1 of Reference 1.

4) Prepare for the next day

$$\tilde{a}_g = \tilde{a}_g + 1440\, \omega_e$$

To obtain the correction for dynamic considerations at each entry to the routine the following operations are performed.

1) The pressure is found in the usual (static) manner.

2) $\cos \psi' = \frac{\bar{y} \cdot \bar{x}}{R}$ $\qquad \bar{x}$ = position vector

3) Find T

4) Interpolate to get the correction to the pressure (DLØG) from Table II.2. 3(b).

$$TRHØA = TRHØA \times 10^{DLØG} \qquad (10)$$

$$P = p\; 10^{DLØG}$$

5) Convert from pressure to density

In the table storage TAB of base values for the region above 90 km log p is stored. The correction for the density is also the correction to the pressure, i.e., the pressure and density are in a 1-1 correspondence and one can work with one of the other and then convert by substituting into the perfect gas law.

$$\rho = \frac{M_o P}{R^* T_m}$$

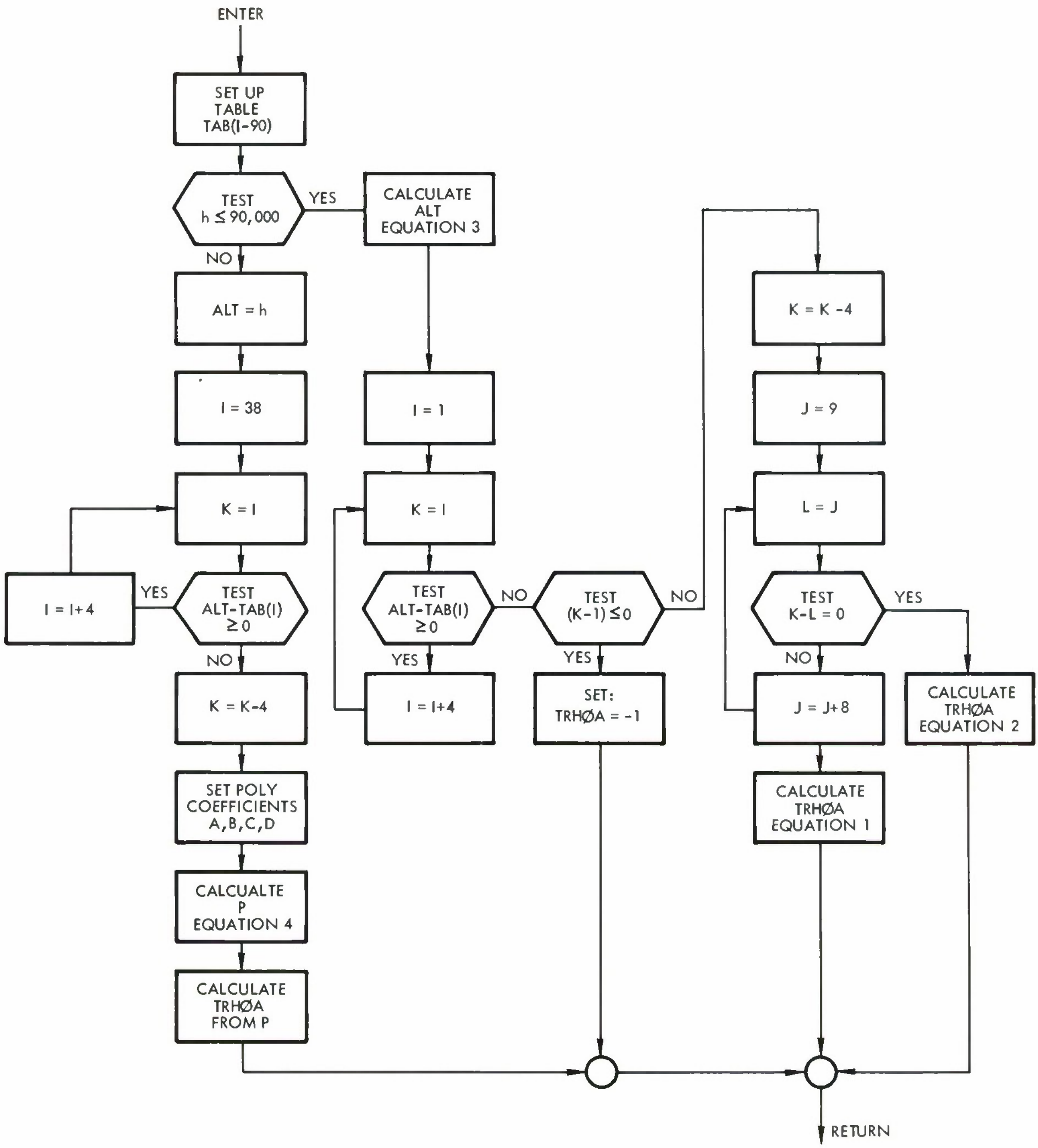


Figure 4-2. CØESA 1962

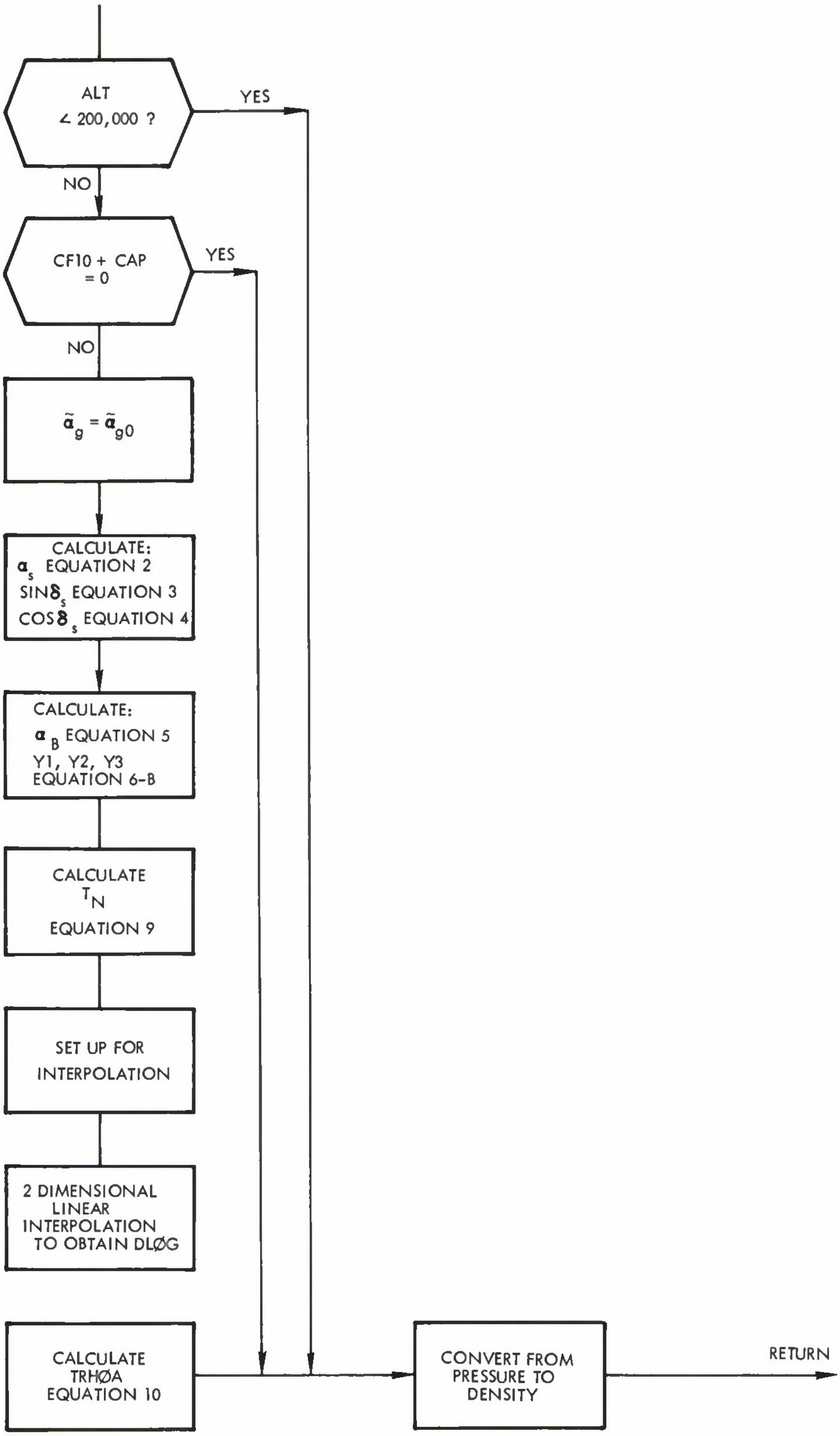


Figure 4-3. Dynamic Consideration Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

CØRMAT

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

APPLY	(ESPØDDC)
UPDATE	(ESPØDEPH)

FUNCTION

Function is to compute the correlation (σ and ρ) matrix given a lower triangular variance-covariance matrix.

USAGE

A. Calling sequence

Call CØRMAT (A, I, B, J, K, L)

B. Input

1. CØMMØN

—

2. Calling sequence

a.	A(I)	A^TA inverse matrix stored at A(I). A(I) is assumed to be a lower triangular matrix, stored by rows. The elements are denoted by a_{ij}.
b.	B(J)	Beginning location of the B matrix (resultant σ and ρ matrix).
c.	K	Dimension of the $(A^TA)^{-1}$ matrix.
d.	L	L = 1, compute normal correlation matrix (i. e., set $b_{ii} = 1$).

C. Output

1. CØMMØN

—

2. Calling sequence

B(J) Correlation matrix starting location.

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRTF

B. Program

—

EQUATIONS

$$b_{ij} = \frac{a_{ij}}{\sqrt{a_{ii}}\,\sqrt{a_{jj}}} \qquad \begin{array}{l} i = 1, \ldots, n \\ j = 1, \ldots, n \end{array}$$

For L = 1

$$b_{ii} = 1 \qquad i = 1, \ldots, n$$

For L = 2

$$b_{ii} = \sqrt{a_{ii}} \qquad i = 1, \ldots, n$$

SUBROUTINE IDENTIFICATION

A. Title

CTØP

B. Segment

ESPØD

ESPØDEPH

C. Called by subroutines

DPRLM	(ESPØD)
MNELTC	(ESPØD)
TPRNT	(ESPØDEPH)

FUNCTION

Function is to convert Cartesian coordinates to polar coordinates.

USAGE

A. Calling sequence

Call CTØP (C, D)

B. Input

1. CØMMØN

—

2. Calling sequence

C(1)	x (e. r.)
C(2)	y (e. r.)
C(3)	z (e. r.)
C(4)	$\dot{x}$ (e. r./min)
C(5)	$\dot{y}$ (e. r./min)
C(6)	$\dot{z}$ (e. r./min)

C. Output

1. CØMMØN

—

2. Calling sequence

D(1)	α (radians)	Right ascension
D(2)	δ (radians)	Declination
D(3)	β (radians)	Flight path angle
D(4)	A (radians)	Azimuth
D(5)	R (e. r.)	Range
D(6)	v (e. r./min)	Velocity

D. Error/action messages

—

SUBROUTINES USED

A. Library

SØRTF

B. Program

ATNQF Arc tangent

EQUATIONS

$$D(1) = \alpha = \tan^{-1}(y/x) \quad 0 \leq \alpha \leq 2\pi$$

$$D(2) = \delta = \tan^{-1}\left[z/\sqrt{x^2 + y^2}\right] \quad -\frac{\pi}{2} \leq \delta \leq \frac{\pi}{2}$$

$$D(3) = \beta = \cos^{-1}\left[(x\dot{x} + y\dot{y} + z\dot{z})/rv\right]$$

$$D(4) = A = \tan^{-1}\left[\frac{r(x\dot{y} - y\dot{x})}{y(y\dot{z} - z\dot{y}) - x(z\dot{x} - x\dot{z})}\right]$$

$$D(5) = r = \sqrt{x^2 + y^2 + z^2}$$

$$D(6) = v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

SUBROUTINE IDENTIFICATION

A. Title

DATE

B. Segment

ESPØDEPH

C. Called by subroutine

TGDJD

FUNCTION

Function is to compute the Gregorian date, given t in minutes from 0^h day of epoch.

USAGE

A. Calling sequence

Call DATE(TMIN)

B. Input

1. CØMMØN

CDAYMN	Number of days in the month
DDAY	Epoch day
DHØUR	Epoch hour
DMIN	Epoch minute
DMNTH	Epoch month
DSEC	Epoch second
DYEAR	Epoch year
TEMP	Temporary storage

2. Calling sequence

TMIN	Minutes from 0^h day of epoch

C. Output

1. CØMMØN

TEMP(3)	Year
TEMP(4)	Month
TEMP(5)	Day
TEMP(6)	Hour
TEMP(7)	Minutes
TEMP(8)	Seconds

2. Calling sequence

—

D. Error/action messages

DATE DATE

SUBROUTINES USED

A. Library

INTF

MØDF

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

DAUX

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

TRAJ
SETIC

FUNCTION

The function is to compute the second derivatives in the equations of motion and control the computation of the second derivatives in the variational equations.

USAGE

A. Calling sequence

Call DAUX

B. Input

1. CØMMØN

TLIST	Numerical integration working storage
TALFA	Constant used in calculating radiation pressure effects
CDAD2M	Drag parameter $C_D A/2m$
CK	Drag parameter K
FLVE	Variational equation control flag
NDPR	Number of Category 1 variable to solve for

2. Calling sequence

—

C. Output

1. CØMMØN

TLIST	Numerical integration working storage
TCRASH	Flag which indicates impact when non-zero

2. Calling sequence

—

D. Error/ action messages

—

SUBROUTINES USED

A. Library

—

B. Program

BØDY

DRAG

PØTENT

MAGN

RPRESS

VAREQ

EQUATIONS

The Cowell formulation of the equations of motion is used:

$$R = \left(x^2 + y^2 + z^2\right)^{1/2}$$

$$\ddot{x} = \frac{-\mu x}{R^3} + \ddot{x}_{\text{bodies}} + \ddot{x}_{\text{drag}} + \ddot{x}_{\text{potential}} + \ddot{x}_{\text{radiation pressure}}$$

$$\ddot{y} = \frac{-\mu y}{R^3} + \ddot{y}_{\text{bodies}} + \ddot{y}_{\text{drag}} + \ddot{y}_{\text{potential}} + \ddot{y}_{\text{radiation pressure}}$$

$$\ddot{z} = \frac{-\mu z}{R^3} + \ddot{z}_{\text{bodies}} + \ddot{z}_{\text{drag}} + \ddot{z}_{\text{potential}} + \ddot{z}_{\text{radiation pressure}}$$

where

$\ddot{x}_{\text{bodies}}$ = The perturbation acceleration due to other bodies in the solar system

$\ddot{x}_{\text{drag}}$ = The perturbation acceleration due to atmosphere drag

$\ddot{x}_{\text{potential}}$ = The perturbation acceleration due to the potential field set by the aspherical earth

$\ddot{x}_{\text{radiation pressure}}$ = The perturbation acceleration due to solar radiation pressure

The tests are made to see which of the above perturbation effects are to be included in the evaluation of the equations of motion.

SUBROUTINE IDENTIFICATION

A. Title

DØN

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

DRAG

FUNCTION

Function is to calculate TDØN, a modifier used in the simulation of the variation of the drag parameter $C_D A/2m$.

USAGE

A. Calling sequence

Call DØN

B. Input

1. CØMMØN

CKSLCT	Flag to indicate whether periodic or secular variation is desired
TLIST	Numerical integration working storage
TALFAG	Right ascension of the Greenwich meridian at midnight day of epoch
CPI	π
TR	Magnitude of the position vector of the vehicle relative to the earth
TEPØCH	Minutes from midnight to epoch

2. Calling sequence

—

C. Output

1. CØMMØN

TDØN	Modifier used in the calculation of the effective drag parameter

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

SIN
SQRT

B. Program

—

EQUATIONS

If CKSLCT = 1

a) Compute the position of the sun.

$$x_s = \cos\left[\alpha_{go} - 180^\circ + \frac{360}{365.25}(t - t_o)\right]$$

$$y_s = \sin\left[\alpha_{go} - 180^\circ + \frac{360}{365.25}(t - t_o)\right] \cos(23.5^\circ)$$

$$z_s = \sin\left[\alpha_{go} - 180^\circ + \frac{360}{365.25}(t - t_o)\right] \sin(23.5^\circ)$$

b) Compute the position of the bulge.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} \cos(30^\circ) & -\sin(30^\circ) & 0 \\ \sin(30^\circ) & \cos(30^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$$

$$\psi = \cos^{-1}\left[\frac{\bar{x}_b \cdot \bar{x}}{R}\right]$$

$$TD\emptyset N = \frac{1}{2}\cos^5\left(\frac{\psi}{2}\right) - \frac{1}{4}$$

If CKSLCT $\neq$ 1

$$TD\emptyset N = \frac{t - t_o}{1440}$$

SUBROUTINE IDENTIFICATION

A. Title

DØT

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

PARØUT	(ESPØDDC)
RPRESS	(ESPØDEPH, ESPØDDC)

FUNCTION

Function is to compute the scalar product C = A · B if D is non-zero; the routine stores the angle between A and B in E.

USAGE

A. Calling sequence

Call DØT (A, B, C, D, E)

B. Input

1. CØMMØN

—

2. Calling sequence

a) A The beginning location of a three-dimensional vector $N = (n_1, n_2, n_3)$

b) B The beginning location of a three-dimensional vector $M = (m_1, m_2, m_3)$

c) D D = 0, do not compute angle between A and B
$D \neq 0$, do compute the angle between A and B

C. Output

1. CØMMØN

—

2. Calling sequence

a) C Scalar product

b) E Angle between A and B

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRTF

B. Programs

ATNQF	Arc tangent
RADSQ	Compute R and R^2 for X, Y, Z

EQUATIONS

$$C = n_1m_1 + n_2m_2 + n_3m_3$$

$$E = \cos^{-1}\left[\frac{C}{|N|\,|M|}\right]$$

SUBROUTINE IDENTIFICATION

A. Title

DPRLM

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

Function is to set up preliminary information for the pre-processor. This information concerns epoch time and mode of epoch position and velocity.

USAGE

A. Calling sequence

Call DPRLM

B. Input

1. CØMMØN

CDEG	Degrees/radian
CJD50	Julian date January 0, 1950
CWE	Earth's rotational rate
DTYPE	Initial conditions type
TEMP	Temporary storage

2. Calling sequence

—

C. Output

1. CØMMØN

DBASE	Number of days from 1950 to day of epoch
TALFAG	α_g for midnight day of epoch
TEPØCH	Epoch time, minutes from midnight
TJDATE	Julian date of midnight, epoch day
TNØMP	Initial spherical coordinates
TNØMX	Initial Cartesian coordinates

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

CTØP	Convert Cartesian to polar coordinates
IPRNT	Prints header page
ITMPCH	Punches the initial epoch time when mean element cards are input
MNELTC	Converts SPADATS mean elements to Cartesian
PIMØD	Takes principle value of angle between 0 and 2π
PTØC	Converts polar to Cartesian
TINIT	Sets up initial time, computes α_g
TMSEP	Modulates initial times and sets up permanent storage

SUBROUTINE IDENTIFICATION

A. Title

DPRØS

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

Function is to issue calls on the sensor and observation loading routines if required.

USAGE

A. Calling sequence

Call DPRØS

B. Input

1. CØMMØN

CLDSTR Cold-start, non-cold, start flag

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

LØDØBS	Control package for loading observation cards from the input tape
LØDSEN	Control package for loading sensors from the input tape
RDCØM	Reads common block from observation tape
REWT	Rewinds observation tape
WRTCØM	Writes common block from observation tape

SUBROUTINE IDENTIFICATION

A. Title

DRAG

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutine

DAUX

FUNCTION

Function is to compute the perturbative acceleration of a vehicle due to atmosphere drag and to account for these effects in the variational equations.

USAGE

A. Calling sequence

Call DRAG

B. Input

1. CØMMØN

FLVE	Variational equation control flag
NDPR	Number of Category 1 variable being solved for
CMTER	Constant = meters per earth radii
CELLIP	Constant = ellipticity of the Earth
TLIST	Numerical integration working storage
TR2	Square of TR
TR	Magnitude of the vector from the center of the Earth to the vehicle
CWE	Constant = rotation rate of the earth (radians/minutes) = ω_e
CDAD2M	Drag parameters $C_D A/2m$
CK	Drag parameter variation K
CKSLCT	Selection flag for periodic or secular variation in drag

2. Calling sequence

—

C. Output

1. CØMMØN

TDRAG	Perturbative acceleration due to drag
VMAT	Matrix of velocity dependent terms in the evaluation of the variational equations
PMAT	Matrix of position dependent terms in the evaluation of the variational equation. (The drag effects are added to the contents of this matrix.)

D. Error/ action messages

—

SUBROUTINES USED

A. Library

SQRT

B. Program

ATMØS

DØN

ØUTER

EQUATIONS

$$R_e = \frac{1 - \epsilon}{\left[1 - \epsilon(2 - \epsilon)\left(\frac{x^2 + y^2}{R^2}\right)\right]^{1/2}} = \text{radius of the Earth}$$

$$\text{Altitude} = R - R_e$$

$$\rho_a = \text{density at the given altitude}$$

$$v_{ax} = \dot{x} + \omega_e y$$

$$v_{ay} = \dot{y} - \omega_e x \qquad \text{Earth-fixed velocity}$$

$$v_{az} = \dot{z}$$

$$v_a = \left(v_{ax}^2 + v_{ax}^2 + v_{ax}^2\right)^{1/2}$$

$$\lambda = \frac{C_d A}{2m} + TD\text{Ø}N \cdot K$$

$$\ddot{z}_{drag} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{ax}$$

$$\ddot{y}_{drag} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{ay}$$

$$\ddot{z}_{drag} = -\rho_a \cdot V_a \cdot \lambda \cdot v_{az}$$

$$
PMAT = PMAT - \lambda \rho_a v_a \begin{bmatrix} 0 & \omega_e & 0 \\ -\omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{\lambda V_a \rho'}{R} \begin{bmatrix} v_{ax}{}^{x} & v_{ax}{}^{y} & v_{ax}{}^{z} \\ v_{ay}{}^{x} & v_{ay}{}^{y} & v_{ay}{}^{z} \\ v_{az}{}^{x} & v_{az}{}^{y} & v_{az}{}^{z} \end{bmatrix}
$$

$$
- \frac{\lambda \rho_a}{V_a} \begin{bmatrix} v_{ax}^2 & v_{ax} v_{ay} & v_{az} v_{az} \\ v_{ax} v_{ay} & v_{ay}^2 & v_{ay} v_{az} \\ v_{ax} v_{az} & v_{ay} v_{az} & v_{az}^2 \end{bmatrix} \begin{bmatrix} 0 & \omega_e & 0 \\ -\omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

$$
VMAT = VMAT - \frac{\lambda \rho_a}{V_a} \begin{bmatrix} v_{ax}^2 & v_{ax} v_{ay} & v_{ax} v_{ax} \\ v_{ax} v_{ay} & v_{ay}^2 & v_{ay} v_{az} \\ v_{ax} v_{az} & v_{ay} v_{az} & v_{az}^2 \end{bmatrix} - \lambda \rho_a v_a \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

SUBROUTINE IDENTIFICATION

A. Title

DRDP

B. Segment

ESPØDDC

C. Called by subroutines

RADR

FUNCTION

Function is to compute the partial of the M^{th} type of observation with respect to the variables, α, δ, β, A, r, v, $C_D A/2m$ and K.

USAGE

A. Calling sequence

Call DRDP (M)

B. Input

1. CØMMØN

NARØW	Starting location where one row of the augmented matrix (A, B) is stored
NDPR	Number of all differential plus initial parameters to solve for (Category 1)
PCMR	Computed slant range
PCSA	Cos A
PRSUB1	R_1 = VR
PSNA	Sin A
PSNE	Sin E
PSTAT	Working storage for sensor information
PUDTI	$\dot{u} = (\dot{u}_1, \dot{u}_2, \dot{u}_3)$
PUI	(u_1, u_2, u_3)
PWDTPP	$\partial\dot{w}/\partial P_i$
PWPP	$\partial w/\partial P_i$

2. Calling sequence

M Observation type number (1, 2, 3, 4, 5, 6)

C. Output

1. CØMMØN

VSTR(NARØW) → VSTR(NARØW + NDPR-1) contains the partial derivatives of the M^{th} type observation with respect to the Category 1 variables being solved for

2. Calling sequence

—

SUBROUTINES USED

A. Library

SQRTF

B. Program

—

EQUATIONS

Range (type 1 observation)

$$\frac{\partial R}{\partial p_i} = u_1 \frac{\partial w_1}{\partial p_i} + u_2 \frac{\partial w_2}{\partial p_i} + u_3 \frac{\partial w_3}{\partial p_i} \qquad p_i > \alpha,\ \delta,\ \beta,\ A,\ \dot{r},\ v,\ C_D A/2m,\ K$$

Azimuth (type 2 observation)

$$\frac{\partial A}{\partial p_i} = \frac{1}{R_1}\left[\frac{\partial w_2}{\partial p_i}\cos A - \left(\frac{\partial w_1}{\partial p_i}\sin\phi^* + \frac{\partial w_3}{\partial p_i}\cos\phi^*\right)\sin A\right]$$

Elevation (type 3 observation

$$\frac{\partial E}{\partial p_i} = \frac{1}{R_1}\left(\frac{\partial w_1}{\partial p_i}\cos\phi^* + \frac{\partial w_3}{\partial p_i}\sin\phi^* - \frac{\partial R}{\partial p_i}\sin E\right)$$

Range Rate (type 4 observation)

$$\frac{\partial \dot{R}}{\partial p_i} = \left(\frac{\partial \overline{w}}{\partial p_i}\cdot\dot{\overline{u}}\right) + \left(\overline{u}\cdot\frac{\partial \dot{\overline{w}}}{\partial p_i}\right)$$

Local Hour Angle (type 5 observation)

$$\frac{\partial H}{\partial p_i} = \frac{1}{R\left(u_1^2 + u_2^2\right)}\left(\frac{\partial w_1}{\partial p_i}u_2 - \frac{\partial w_2}{\partial p_i}u_1\right)$$

Declination (type 6 observation)

$$\frac{\partial D}{\partial p_i} = \frac{\dfrac{\partial w_3}{\partial p_i} - \dfrac{\partial R}{\partial p_i}\sin D}{R\cos D}$$

SUBROUTINE IDENTIFICATION

A. Title

DRIVER

B. Segment

ESPØD

C. Called by subroutine

—

FUNCTION

The ESPØD main control serves as the coordinator of all activities involving the three segments ESPØD, ESPØDDC, and ESPØDEPH. It utilizes existing EXECMØD1 and EXECMØD2 routines for pulling these segments off the master tape when they are needed. Control is always returned to this routine when any of the three segments have completed their job.

USAGE

A. Calling sequence

—

B. Input

1. CØMMØN

CLDSTR	Cold-start, non-cold start flag
CØNVR	Flag to indicate if previous case converged or diverged
DCFLG	ESPØDDC control flags
DNREV	Control cells for seven-card input
FGELEM	Flag to indicate element cards read
FGICØN	Flag to indicate ICØND card read
IFTEX	Indicates mode of exit from FIT
NARØW	Starting location where one row of the augmented matrix (A, B) is stored
NATA	Starting location of where the triangular $A^T A$ is stored

NBDNS	Starting location for the bounds used by LEGS
NDPAR1 NDPAR2 NDPAR3 NDPAR4	Starting location where the four sets of solution vectors will be stored
NDPR	Number of all differential plus initial parameters to solve for (Category 1)
NDTCT	Δt, t table counter
N1CPR	Number of initial conditions parameters to solve for
N1DENT	Number of entries in the N1DLED list
N1DLED	Starting location of where the observation deletion table begins
N1DP	1dentifier for table indicating Category 1 type variables to be solved for (refer to 1VSTR)
N1TCT	1teration counter
N1TER	Maximum number of iterations
NMBER	Number of observations
NØEPØC	Flag to indicate epoch not established
NPAR	Starting location for the parameter list
NPBIS	Identifies table for current estimates of Category 2 variables
NPR	Number of all parameters to solve for
NPRCD	1dentifies table for definition of Category 2 variables to be solved for (refer to 1VSTR)
NR	Starting location of where the inverse A^TA is stored
NRTMP	Starting location of temporary storage for special handling of the R matrix
NSCALE	Identifies the starting location of where the scaling factors are stored
NSMAT	Identifies the starting location of a priori S matrix

NSSTB	Identifies the starting location where station information concerning computed sigmas and means of residuals are stored
NSTAT	Identifies the starting location of the master sensor table
NUBS	Identifies the starting location of the observation table
N1 N2 N3	Counters for geopotential routine
PREFLG	ESPØD control flags
PSTFLG	ESPØDEPH control flags

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

PANT

B. Program

ASSIGN	Storage assignments for VSTR and IVSTR
DPRLM	Sets up preliminary information
DPRØS	Drives the sensor and observation loading routines
ELMØD	Searches the SEAI tape for the orbital elements
JDCSRCH	Searches the input data for the JDC card
READPR	Preliminary data read routine

REWT	Rewinds the observation tape
SDELET	Moves input storage into variable storage
SETCØN	Sets the program constants
SETTAB	Sets up variable storage
SNØMIC	To move the initial conditions from variable storage into TNØMX, TNØMP or TMNEL
STSMAT	To convert the upper triangular S matrix from human to machine units, then move it to variable storage
SUPMAT	To move the initial update matrix from temporary to permanent storage
WRTCØM	To write CØMMØN data storage on the work tape

SUBROUTINE IDENTIFICATION

A. Title

DYNAT

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

ATMØS

FUNCTION

The function is to determine the density of the atmosphere in the range from 0-1600 km employing ARDC 59 and the Paetzold dynamic atmosphere.

USAGE

A. Calling sequence

Call DYNAT

B. Input

1. CØMMØN

ALT	Table for Paetzold dynamic atmosphere
TALT	Altitude, meters
CAP	A_p for this time
CF10	F_{10} for this time
PHIH	Table for Paetzold dynamic atmosphere
PSTAR	Table for Paetzold dynamic atmosphere
TALFAG	Sidereal time at 0^h day of epoch
THETH	Table for Paetzold dynamic atmosphere
TLIST(2)	Time in minutes from 0^h day of epoch
TLIST(4)	x
TLIST(5)	y
TLIST(6)	z

2. Calling sequence

—

C. Output

1. CØMMØN

TRHØA Density, kg/m^3

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

SINF

CØSF

B. Program

ATM59	Static atmosphere (ARDC 1959 model)
PØLY	Evaluates N^{th} order polynomial
APF10	Gets values of A_p and F_{10} for time found in TLIST(2)

EQUATIONS

The right ascension of Greenwich, α_g, is input for time of epoch and is increased by a degree once per day through a test on t - TE $\geq$ 1440 minutes, where TE is increased 1440 each time α_g is increased. The right ascension of Greenwich effectively gives the orientation of the earth for the date in question.

Auxiliary quantities a, b and c are computed for interpolations in altitude for the Paetzold model, where h is altitude in km.

$$a = \frac{x_1}{2}(x_1 - 1)$$

$$b = -(x_1 - 1)(x_1 + 1)$$

$$c = \frac{x_1}{2}(x_1 + 1)$$

Here x_1 is a function of h only and is computed according to a series of tests on h in the interpolation routine.

A table is set up for angles and density related to altitude as follows:

$$m = g(a) + (220 - F)\left[0.0060 - 0.002g(a)\right]$$

$$\log \rho^*(h) = \log \rho_3(h) - i(220, h)\frac{220 - F}{F} + k(200, h)\frac{A_p}{200} - a(220, h)m$$

$$\psi(h) = i(220, h)\frac{200 - F}{F}\theta_i(h)$$

$$\theta(h) = \theta_s(h) - \Delta_1(h)\frac{i(220, h)\frac{220 - F}{F} + ma(220, h)}{i(220, h) + a(220, h)} - \Delta_2\theta(h)\frac{220 - F}{F}$$

where F is an input quantity and g(a) is a polynomial related to season through a test on the month, D

$$D = \frac{a_g - 99}{30 \times 0.985} + 1$$

which dictates one set of coefficients for $D < 7$ and another for $D > 7$. For a given h, ρ^*, ψ, and θ are interpolated from the table using a, b, c.

The density is then computed as follows:

$h > 150$ km

$$r = \sqrt{x^2 + y^2}$$

$$\cos\theta = \frac{x\cos a_g + y\sin a_g}{r}$$

$$\sin\theta = \frac{-x\sin a_g + y\cos a_g}{r}$$

$$\sin 2\theta = 2\sin\theta\cos\theta$$

$$\cos 2\theta = \cos^2\theta - \sin^2\theta$$

$$i(\theta) = A_o + B_1 \sin\theta + C_1 \cos\theta + B_2 \sin 2\theta + C_2 \cos 2\theta$$

$$f(\theta) = B_o + B_1 \sin\theta + C_1 \cos\theta + B_2 \sin 2\theta + C_2 \cos 2\theta$$

$$\log \rho(h) = \log \rho^*(h) - \psi(h) i(\theta) - \theta(h) f(\theta)$$

$h < 130$ km

ARDC 1959 $> \rho(h)$

$130 < h < 150$ km

$$\rho = \frac{1}{400}\left\{\left[400 - (h - 130)^2\right]\rho_{ARDC} + (h - 130)^2 \rho_{Paet}\right\}$$

$$\rho_{Paet} = e^{\log \rho(h)}$$

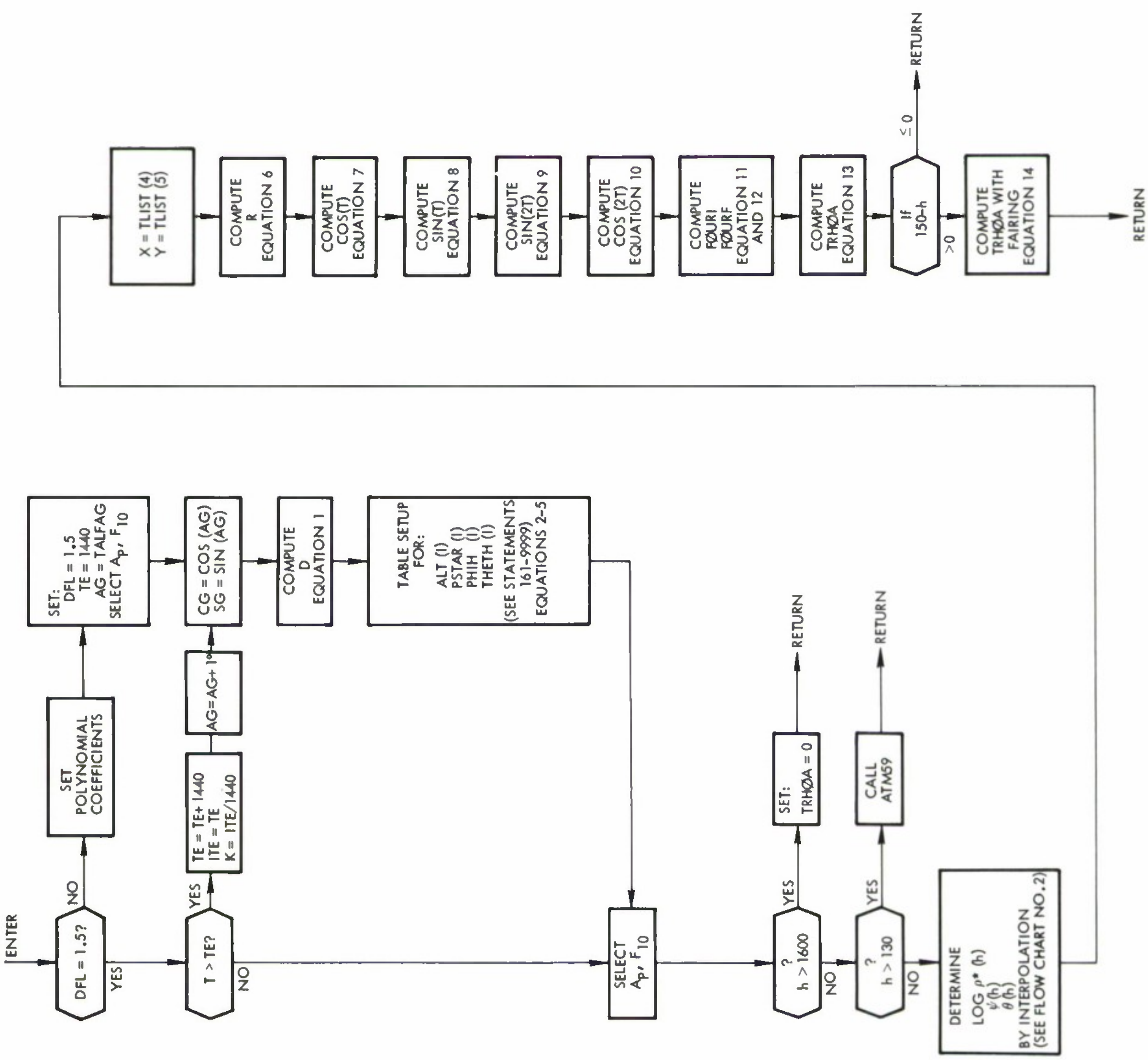


Figure 4-4. DYNAT Subroutine Flow Diagram

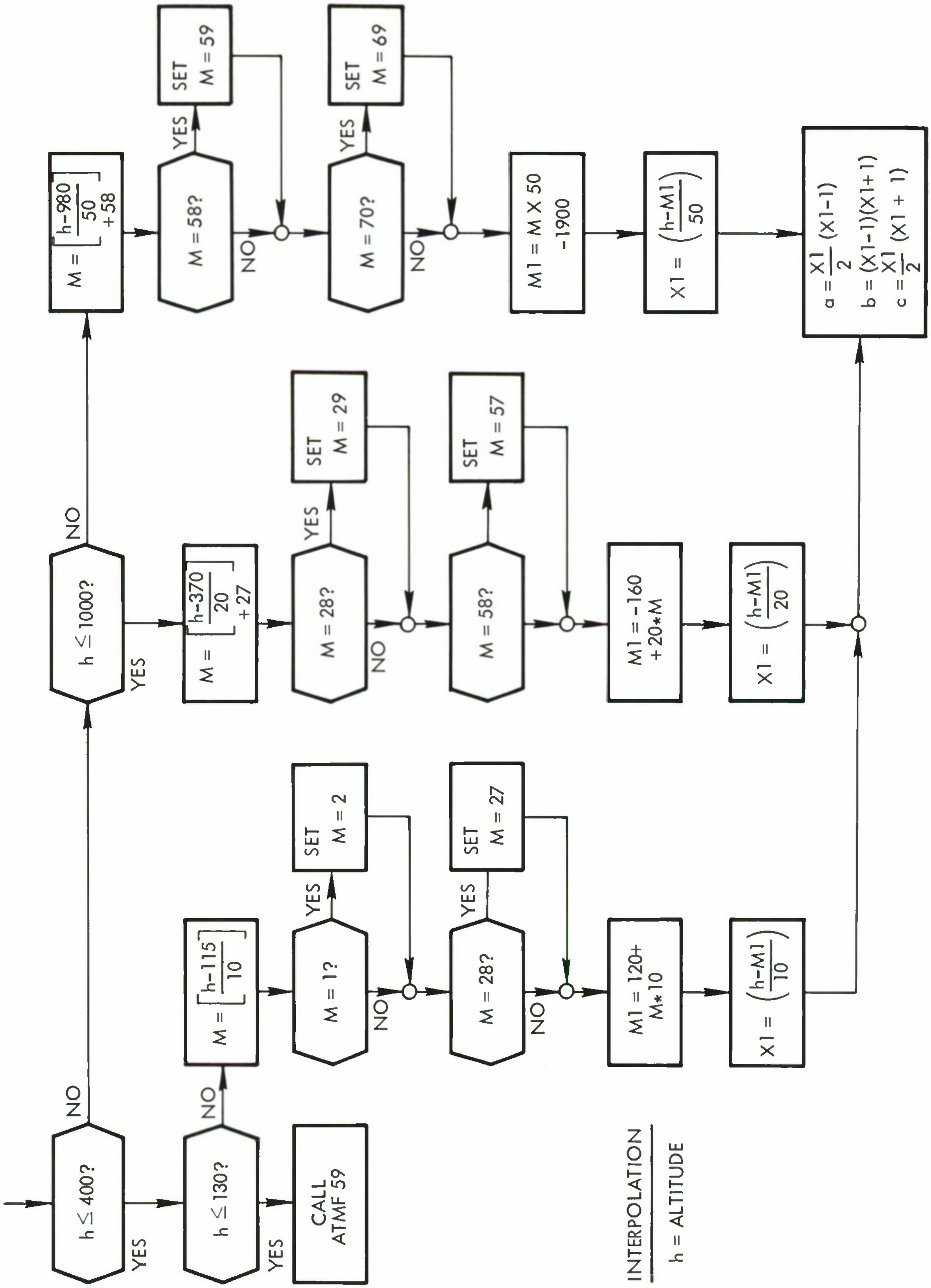


Figure 4-5. Altitude Interpolation Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

ELMLØD

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

This subroutine searches the SEAI tape for the orbital elements (initial conditions) for the satellite number in cell DVEHN. If no match is found, a command occurs on and off-line.

USAGE

A. Calling sequence

Call ELMLØD

B. Input

1. CØMMØN

DVEHN — Vehicle number and name (BSD)

2. Calling sequence

—

C. Output

1. CØMMØN

DNREV(1) — Type of input flag

DNREV(1) = 1 (T is input, flag)
DNREV(1) = 2 (ΔN is input, flag)
DNREV(1) = 3 (N is input, flag)

DNREV(2) — Input, number

DNREV(2) = T input
DNREV(2) = ΔN input
DNREV(2) = N input

DNREV(3)	Epoch revolution number
DSDAY	Epoch time (days and fractions of days from 1950)
DYEAR	Year of epoch (2 digits)
TMNEL(1)	No, epoch revolution number
(2)	axN $\underline{N}$ component of $\underline{a}$
(3)	aYN $\underline{\overline{M}}$ component of $\underline{\overline{a}}$
(4)	hX
(5)	hY, components of angular momentum per unit mass
(6)	hZ
(7)	Lo, mean longitude (rad)
(8)	Co, rate of change of period (days/rev^2)

D. Error/action messages

1. Off-line comment:

   "ELEMENTS FOR SATT.____NOT ON SEAI."

2. On-line comments:

   "TAPE 04 BAD - MOUNT BACKUP"
   "ELEMENTS FOR SATT.____NOT ON SEAI.
   TYPE - GO TO REREAD $\overline{\text{SEAI}}$, STOP FOR NEXT CASE."

3. Action:

   SUBROUTINE ERRØR

SUBROUTINES USED

A. Library

GLØP
READT
STARTGØ
STARTRD
TAPCK
TAPEØUT
ZCHEK

B. Program

ERRØR	Error routine
FLEX	Flexowriter print routine

SUBROUTINE IDENTIFICATION

A. Title

ERRØR

B. Segment

ESPØD
ESPØDEPH
ESPØDDC

C. Called by subroutine

GENERAL PURPOSE ROUTINE

FUNCTION

This is a general error subroutine. Cell 3 is set in ESPØD main control by the following instructions: TJM, ERRØR. ØLØ1; JMP, ERRØR. The contents of the A, Q, and JA register are printed. Control is returned to ESPØD to start the next case.

USAGE

A. Calling sequence

Call ERRØR

B. Input

1. CØMMØN

—

2 Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

1. Off-line comment:

"SUBERR EXIT OCCURRED, SUBERR J LOC, PROGRAM JMP LOC, A REGISTER, Q REGISTER"

2. Action:

Go to next case.

SUBROUTINES USED

A. Library

GLØP
ØCTØR
RPLLØD
RTETYPE

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

EXIT

B. Segment

ESPØD

C. Called by subroutine

MNELTC

FUNCTION

The function is to empty the output buffers and go to the next case.

USAGE

A. Calling sequence

Call EXIT

B. Input

1. CØMMØN

—

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

FIT

B. Segment

ESPØDDC

C. Called by subroutine

INTEG

FUNCTION

This subroutine monitors the flow of information throught the following sequence of events:

a) Asking if this iteration is converging or diverging

b) Writing of CØMMØN onto the tape 7 if converging

c) Forming the solution vector of the differential correction and applying it to give new estimates of the parameters being solved for

d) Setting the bounds for the next iteration

e) Punching current estimates of parameters being solved for

f) Writing of new elements on tape 7.

USAGE

A. Calling sequence

Call FIT

B. Input

1. CØMMØN

IFTEX	Indicates mode of exit from FIT
NDPAR1	Starting location where the solution vector will be stored
NITCT	Iteration counter
NITER	Number of entries in the NIDLED list
NPR	Number of all parameters to solve for

PØBCNT	Number of observations actually included on any iteration
TEMP	Temporary storage
TSUS	Current RMS
TSUSB	Best RMS so far
TSUSP	Predicted RMS for next iteration
TZ	Indicates if solution was affected by bounds
VSTR	Variable storage
XBSQ	Scale factor for BNDS to cause subsequent solutions to be affected by bounds
CFTEPS	ϵ for convergence criterion
KØUT	Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

VSTR (NBDNS) Array in variable storage containing the set of bounds to be used on the next iteration

2. Calling sequence

—

D. Error/action messages

* * * MAJOR PROGRAM ERROR, . . . , POSSIBLE INPUT AND/OR MACHINE ERROR

This message is printed when the total RMS on the first iteration exceeds the maximum floating point number the machine can handle. The action taken is to return and begin processing the next case.

SUBROUTINES USED

A. Library

SQRTF

GLØP

B. Program

APPLY	Applies DC solution vector and prints
BØUNDS	Scales bounds with a scale factor
EXIT	Empties output buffers and goes to next case
LEGS2	Lest squares package, solves AX = B
NPRPCH	Punches ICØND, BISES, BNDS values at the end of each iteration
REWT	Rewinds observation tape
WRTCØM	Writes CØMMØN block from observation tape

SUBROUTINE IDENTIFICATION

A. Title

GPERT

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

PØTENT

FUNCTION

The function of this subroutine is computing the perturbative acceleration of a spacecraft resulting from the fact that the Earth is not a homogeneous sphere. (The resulting harmonics are termed zonal, sectorial, and tesseral.)

USAGE

A. Calling sequence

Call GPERT

B. Input

1. CØMMØN

SIPH	Sin ϕ where ϕ is the geocentric latitude of the vehicle
CØPH	Cos ϕ
SILA	Sin λ where λ is the east longitude of the vehicle
CØLA	Cos λ
SNALF	Sin α where α is the right ascension of the vehicle
CSALF	Cos α
FJ	Twelve cell array containing the values of the desired zonal harmonic constants
C	Six by six array used in the simulation of the sectorial and tesseral harmonics (see JCS subroutine)
S	Six by six array as above

N1 Degree of the highest zonal harmonic

N2 Degree of the highest sectorial harmonic

N3 Degree of the highest tesseral harmonic

CMU Earth's GM

TR Magnitude of the radius vector, Earth to vehicle

TR3 The cube of TR

2. Calling sequence

—

C. Output

1. CØMMØN

TPØT Perturbative acceleration of the vehicle in x, y, z, inertial coordinate system due to earth's potential function

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

This is a recursive computation, formulated as described in the following paragraphs.

Acceleration in a local rectangular system (f, g, h) with h along the outward geocentric vertical, f directed south and g directed east.

$$a_f = \cos\phi \sum_{n=2}^{N1} \left(J_n r^{-n-2}\right) \rho'_n$$

$$+ \sum_{m=2}^{N2} m r^{-m-2} \sin\phi \left(\sec\phi\ \rho_m^m\right)(C_{mm} \cos m\lambda + S_{mm} \sin m\lambda)$$

$$- \sum_{m=1}^{N3} \sum_{n=m+1}^{N3} r^{-n-2} \left(\cos\phi\, \rho_n^{m'}\right)(C_{nm} \cos m\lambda + S_{nm} \sin m\lambda)$$

$$a_g = - \sum_{m=2}^{N2} m r^{-m-2} \left(\sec\phi\ \rho_m^m\right)(C_{mm} \sin m\lambda - S_{mm} \cos m\lambda)$$

$$- \sum_{m=1}^{N3} m \sum_{n=m+1}^{N3} r^{-n-2} \left(\sec\phi\, \rho_n^m\right)(C_{nm} \sin m\lambda - S_{nm} \cos m\lambda)$$

$$a_h = \sum_{n=2}^{N1} (n+1) \left(J_n r^{-n-2}\right) \rho_n$$

$$- \cos\phi \left[\sum_{m=2}^{N2} (m+1) r^{-m-2} \left(\sec\phi\ \rho_m^m\right)(C_{mm} \cos m + S_{mm} \sin m\lambda) \right.$$

$$\left. + \sum_{m=1}^{N3} \sum_{n=m+1}^{N3} (n+1) r^{-n-2} \left(\sec\phi\, \rho_n^m\right)(C_{nm} \cos m\lambda + S_{nm} \sin m\lambda) \right]$$

where

$$\rho_n = \left[(2n-1) \sin\phi\, \rho_{n-1} - (n-1)\, \rho_{n-2}\right]/n$$

$$\rho_o = 1$$

$$\rho_1 = \sin\phi$$

$$\rho'_n = \sin\phi\, \rho'_{n-1} + n\, \rho_{n-1}$$

$$\rho'_1 = 1$$

and

$$\left(\sec\phi\ \rho_m^m\right) = (2m - 1)\cos\phi\left(\sec\phi\ \rho_{m-1}^{m-1}\right)$$

$$(\sec\phi\ \rho_1^{1}) = 1$$

$$\sec\phi\ \rho_n^m = \left[(2n - 1)\sin\phi\left(\sec\phi\ \rho_{n-1}^m\right) - (n + m - 1)\left(\sec\phi\ \rho_{n-2}^m\right)\right]/(n-m)$$

$$\sec\phi\ \rho_{m-1}^m = 0$$

and

$$\left(\cos\phi\ \rho_m^{m'}\right) = -m\sin\phi\left(\sec\phi\ \rho_m^m\right)$$

$$\left(\cos\phi\ \rho_m^{m'}\right) = -n\sin\phi\left(\sec\phi\ \rho_n^m\right) + (n + m)\left(\sec\phi\ \rho_{n-1}^m\right)$$

These accelerations are then rotated to an x, y, z inertial system and scaled by the Earth's GM(μ)

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \mu \begin{bmatrix} \cos\alpha\sin\phi & -\sin\alpha & \cos\alpha\cos\phi \\ \sin\alpha\sin\phi & \cos\alpha & \sin\alpha\cos\phi \\ -\cos\phi & 0 & \sin\phi \end{bmatrix} \begin{bmatrix} a_f \\ a_g \\ a_h \end{bmatrix}$$

SUBROUTINE IDENTIFICATION

A. Title

HUMAH

B. Segment

ESPØD
ESPØDDC
ESPØDEPH

C. Called by subroutines

APPLY	(ESPØDDC)
MATPCH	(ESPØDDC)
NPRPCH	(ESPØDDC)
UPDATE	(ESPØDEPH)
STSMAT	(ESPØD)
SUPMAT	(ESPØD)

FUNCTION

This subroutine functions in converting a vector, $A^T A$ matrix, or the $(A^T A)^{-1}$ matrix from machine units to human units or from human units to machine units. The $A^T A$ is an upper triangle matrix and the $(A^T A)^{-1}$ is a lower triangular matrix.

USAGE

A. Calling sequence

Call HUMAH (A, I, B, J, K, L)

B. Input

1. CØMMØN

—

2. Calling sequence

a) A(I) Starting location of the array to be converted

b) B(J) Starting location of the scaling vector

c) K Dimension of A and B

d) L L = +1, if a vector is to be converted from machine units to human units

L = -1, if a vector is to be converted from human units to machine units

L = +2, if an $A^T A$ matrix is to be converted from machine units to human units

L = -2, if an $A^T A$ matrix is to be converted from human units to machine units

L = +3, if an $(A^T A)^{-1}$ matrix is to be converted from machine units to human units.

L = -3, if an $(A^T A)^{-1}$ matrix is to be converted from human units to machine units

C. Output

1. CØMMØN

—

2. Calling sequence

A(I) The matrix or vector A in the changed units defined by L

D. Error/ action messages

—

SUBROUTINES USED

A. Library

XABSF

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

IDSUB

B. Segment

ESPØD

C. Called by subroutines

BCDØBS
ØBSLØD
READPR
SENRD
SNSGET

FUNCTION

This subroutine replaces leading blanks of four character sensor numbers with zeros. Enter this subroutine with the four characters right adjusted in the A register and exit with the ID in the A register.

USAGE

A. Calling sequence

Call IDSUB (A)

B. Input

1. CØMMØN

—

2. Calling sequence

A Contains sensor number to be checked

C. Output

1. CØMMØN

—

2. Calling sequence

A Leading blanks of A replaced by zeros

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

INTEG

B. Segment

ESPØDDC

C. Called by subroutines

ESPØD

FUNCTION

This subroutine controls the logical flow of information through the differential correction package (ESPØDDC).

USAGE

A. Calling sequence

Call INTEG (EXIT)

B. Input

1. CØMMØN

CØUNT	Lines counter
DCFLG	Dc package control flags
IFTEX	Indicated mode of exit from FIT
PSIG	Sigma list
TALT	Altitude, meters
TG	Time to integrate to
TRHØA	Density, kg/m^3
MT	Observation tape number
IØUT	Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

EXIT	Gives the status (convergence or divergence) of the differential correction (see IFIT)

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

FIT	Logic control for dc options
PARSET	Initialize partials package
PRSSTB	Compute and print residual
RADR	Driver for partials package
RDCØM	Reads CØMMØN block from observation tape
REWT	Rewinds observation tape
SELECT	Select next time to integrate to
SETIC	Initialize integration list
TPRLM	Sets up data for integration program
TRAJ	Driver for integration program
WRTCØM	Writes CØMMØN block on observation tape
REJECT	Computes final values of RMS by observation type from accepted observations of the last pass

SUBROUTINE IDENTIFICATION

A. Title

INTPL

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

BØDY (ESPØDDC, ESPØDEPH)
RPRESS (ESPØDDC, ESPØDEPH)

FUNCTION

The function of this subroutine is to read the ephemeris tape for the positions of the sun and moon or for the positions and velocities of the sun, moon, Mars, Venus, Jupiter, and Saturn with respect to the Earth and to rotate these coordinates to true of midnight day of epoch.

USAGE

A. Calling sequence

Call INTPL (VEL, T, BASE)

B. Input

1. CØMMØN

CJD50	Julian date of 1950.0
CKMER	Kilometers per Earth radii constant

2. Calling sequence

VEL	Flag: if 0, position of sun, moon only; if 1, positions and velocities as described above.
T	Time in minutes from midnight day of epoch
BASE	Days from 1950.0 to midnight day of epoch

C. Output

1. CØMMØN

XN	Positions of Earth, moon, sun, Venus, Mars, Saturn, and Jupiter referenced to the Earth
XNDØT	Velocities of above bodies with respect to the Earth

2. Calling sequence

—

D. Error/action messages

1. If the time requested is after December 19, 1969, the following message is printed off-line:

   * * * EPHEMERIS TAPE ARGUMENT TOO LARGE . . .
   T = + . XXXXXXXXXX XX SECONDS FROM 1950. 0

   and the program goes onto the next case through the subroutine ERRØR.

2. If the time requested is before August 28, 1960, the following message is printed off-line:

   * * * EPHEMERIS TAPE ARGUMENT TOO SMALL . . .
   T = + . XXXXXXXXXX SECONDS FROM 1950. 0

   and the program goes onto the next case through the subroutine ERRØR

3. If an end-of-file is encountered while reading the ephemeris tape, the following message is printed off-line:

   * * * END OF FILE ENCOUNTERED READING THE EPHEMERIS TAPE

   and the program goes onto the next case through the subroutine ERRØR.

SUBROUTINES USED

A. Library

—

B. Program

INTR
RØTRU
GLØP
STØPGØ
FLEX

EQUATIONS

None

SUBROUTINE IDENTIFICATION

A. Title

IPRNT

B. Segment

ESPØD

C. Called by subroutines

DPRLM
LØDØBS

FUNCTION

The function is to print out the header, initial conditions, vehicle number and name. If I = 1, the routing gives the normal output (i. e., header, initial conditions, vehicle number and name). If I = 2, "NEW EPOCH" is printed with only the epoch time, and the initial conditions are included.

USAGE

A. Calling sequence

Call IPRNT (I)

B. Input

1. CØMMØN

CDAD2M	$C_DA/2m$
CDEG	Degrees/radian
CLDSTR	Cold-start, non-cold-start flag
DDAY	Epoch day
DHEAD	Header from JDC card
DHØUR	Epoch hour
DMIN	Epoch minute
DMNTH	Epoch month
DSEC	Epoch second
DVEHN	Vehicle number and name
DYEAR	Epoch year
HEADER	Contents of REM card
PREFLG	Preprocessor control flags
TALFAG	α_g for midnight day of epoch
TEMP	Temporary storage
TNØMP	Initial spherical coordinates
TNØMX	Initial cartesian coordinates

2. Calling sequence

I = 1, normal output
I = 2, print "NEW EPOCH"

C. Output

1. CØMMØN

——

2. Calling sequence

——

D. Error/action messages

SUBROUTINES USED

A. Library

GLØP
PANT

B. Program

PRCONS Prints the program constants

SUBROUTINE IDENTIFICATION

A. Title

ITMPCH

B. Segment

ESPØD

C. Called by subroutine

DPRLM

FUNCTION

This subroutine punches the ICTYP = 1.0 and the ITIME cards. It is entered only when SPADATS seven-card element sets are input or elements are obtained from the SEAI tape.

USAGE

A. Calling sequence

Call ITMPCH

B. Input

1. CØMMØN

DDAY	Epoch day
DHØUR	Epoch hour
DMIN	Epoch minute
DMNTH	Epoch month
DSEC	Epoch second
DTYPE	Initial conditions type
DYEAR	Epoch year

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library
GLØP

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

JCS

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

TPRLM

FUNCTION

This subroutine sets up working storage for simulation of zonal, sectorial, and tesseral harmonics of the Earth's potential function.

USAGE

A. Calling sequence

Call JCS

B. Input

1. CØMMØN

ZØNAL	Array of 11 flags, non-zero to include the desired harmonic ($J_2, J_3, \ldots, J_{12}$)
SECT	Array of five flags, non-zero to include the desired sectorial harmonic ($J_2^2, J_3^3, \ldots, J_6^6$)
TESS	Array of code words for selection of tesseral harmonics, where each cell is of the form N * 10 + M where N is the degree and M the order of the desired tesseral
CJ	Twelve cell array containing the values of $J_2, J_3, \ldots, J_{12}$
CJNM	Six by six array containing $J_1^1, J_2^2, \ldots, J_6^6$ along the main diagonal, $J_2^1, J_3^1, \ldots, J_6^5$ below the diagonal and $\lambda_2^1, \lambda_3^1, \ldots, \lambda_6^5$ above the diagonal
CLAMNN	Five cell array containing $\lambda_2^2, \lambda_3^3, \ldots, \lambda_6^6$
CDEG	Degrees per radian constant

2. Calling sequence

—

C. Output

1. CØMMØN

FJ	Twelve cell array which contains 0, J_2 or 0, J, or 0, . . . , J_{12} or 0
C	Six by six array used in simulation of sectorial and tesseral harmonics
S	Six by six array used in simulation of sectorial and tesseral harmonics
N1	Degree of largest zonal harmonic requested
N2	Degree of largest sectorial harmonic requested
N3	Degree of largest tesseral harmonic requested

2. Calling sequence

—

D. Error/action messages

If the order of a requested tesseral is greater than or equal to the degree, the following message is printed off-line:

* * * ILLEGAL TESSERAL NM REQUESTED, THE PROGRAM IS IGNORING IT AND PRECEEDING.

SUBROUTINES USED

A. Library

CØS
SIN

B. Program

—

EQUATIONS

$$C_{n,m} = J_{n,m} \cos (m \lambda\ n,m)$$

$$S_{n,m} = J_{n,m} \sin (m \lambda\ n,m)$$

SUBROUTINE IDENTIFICATION

A. Title

JDCSRCH

B. Segment

ESPØD

C. Called by subroutine

READPR (ESPØD)

FUNCTION

This subroutine searches the input data for a JDC card. When a JDC card is found, flags are set for READPR and for the control of the entire orbit determination program.

USAGE

A. Calling sequence

CALL JDCSRCH

B. Input

CARBUF

C. Output

1. CØMMØN

DATA, DATA + 1200 FGICØN FGITIM FGICTY FGELEM FGCAT1 FGCAT2 FGBNDS FGDELE FGAUX	set to 0.0
DVEHN DHEAD CLDSTR PREFLG DCFLG PSTFLG	set according to JDC card

D. Error/action messages

Message	Meaning and Action
NO JDC IN 400 CARDS	No JDC card could be found in 400 input data cards. Job is terminated by transfer of control to READPR · GETØFF.
JDC CARD NOT FOUND. ID WHICH TERMINATED RUN IS ΔΔΔAAAAA	A identification name (AAAAA) was found to be either ENDAT, ZZZZZ, or EEEEE. Job is terminated by transfer of control to READPR · GETØFF.

SUBROUTINES USED

A. Library

—

B. Program

RDCØM	Read in COMMON if a non-cold start
PANT	Eject a page for next case
GLØP	Place error messages on output tape
FLEX	Type error message on console typewriter
READPR·RDØNE	Read one card and print it
XSRCH	Read the JDC card in proper format

CROSS REFERENCES

READPR·CARDIM
READPR·ID1
READPR·ID2
READPR·DC8
READPR·DC9
READPR·DC10
READPR·DC11
READPR·DC12
READPR·GETØFF

SUBROUTINE IDENTIFICATION

A. Title

LEGS1

B. Segment

ESPØDDC

C. Called by subroutine

RADR

FUNCTION

This subroutine transforms the augmented matrix (A, B) of the system Ax = B into the augmented normal matrix.

$$\begin{bmatrix} A^T A & A^T B \\ B^T A & B^T B \end{bmatrix}$$

Since the augmented normal matrix is symmetric, only the upper triangle part is stored. Also, if a row is deleted, the count, PØBCNT, is reduced by 1.

USAGE

A. Calling sequence

Call LEGS1 (K, I3, SUS)

B. Input

1. CØMMØN

IVSTR	Fixed point variable storage
NARØW	Identifies the starting location where 1 row of the augmented matrix (A, B) is stored
NATA	Identifies the starting location of where the triangular $A^T A$ is stored
NBDNS	Identifies the starting location for the bounds, used by LEGS2
NPR	Number of all parameters to solve for

NIDLED Identifies the starting location of where the observation deletion table begins

PØBCNT Number of observations actually included on any one iteration

2. Calling sequence

K Row number of A

I3 I3 is used only when K = 1. If $I3 \geq 0$, the $A^T A$ section is cleared before computing $A^T A$. If $I3 < 0$, the section is not cleared.

C. Output

1. CØMMØN

VSTR (NATA) Where the triangular $A^T A$ is stored

2. Calling sequence

SUS Current sum of squares of weighted residuals

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

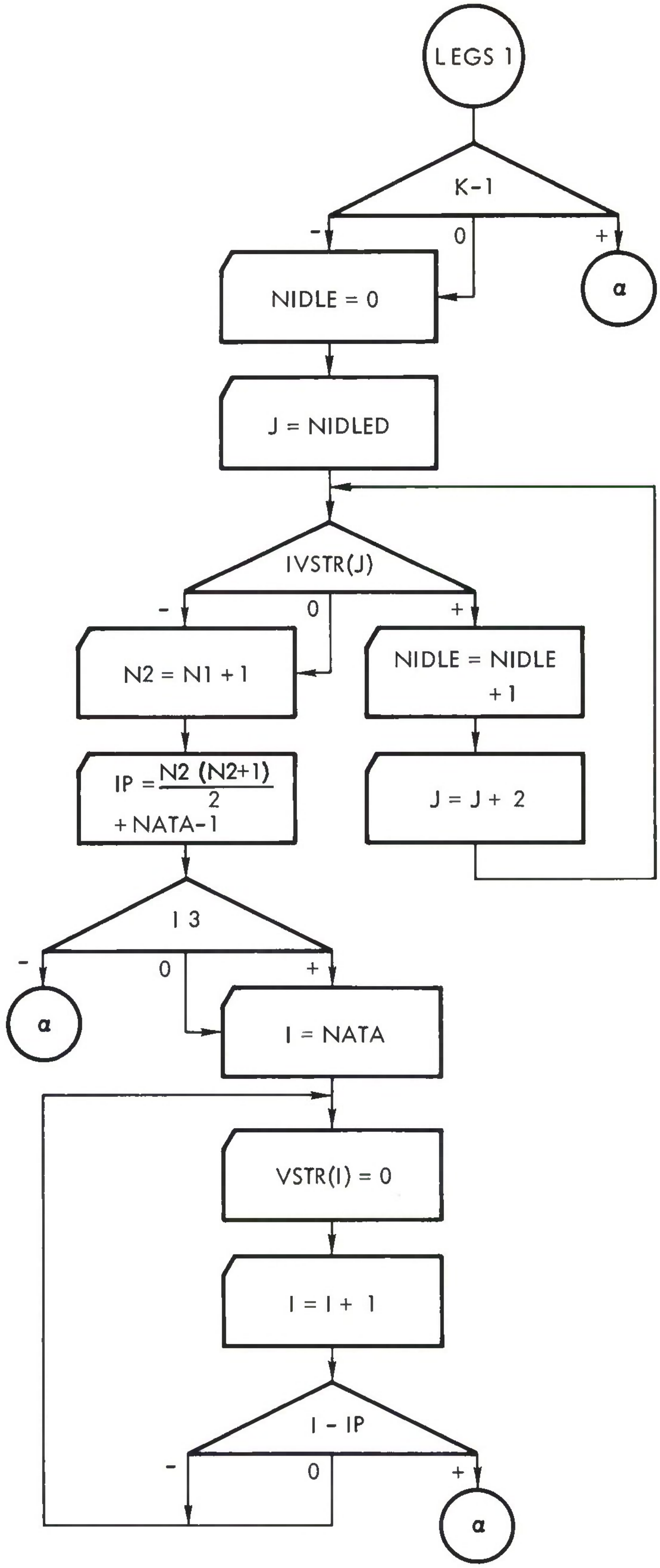


Figure 4-6 a. LEGS1 Flow Diagram

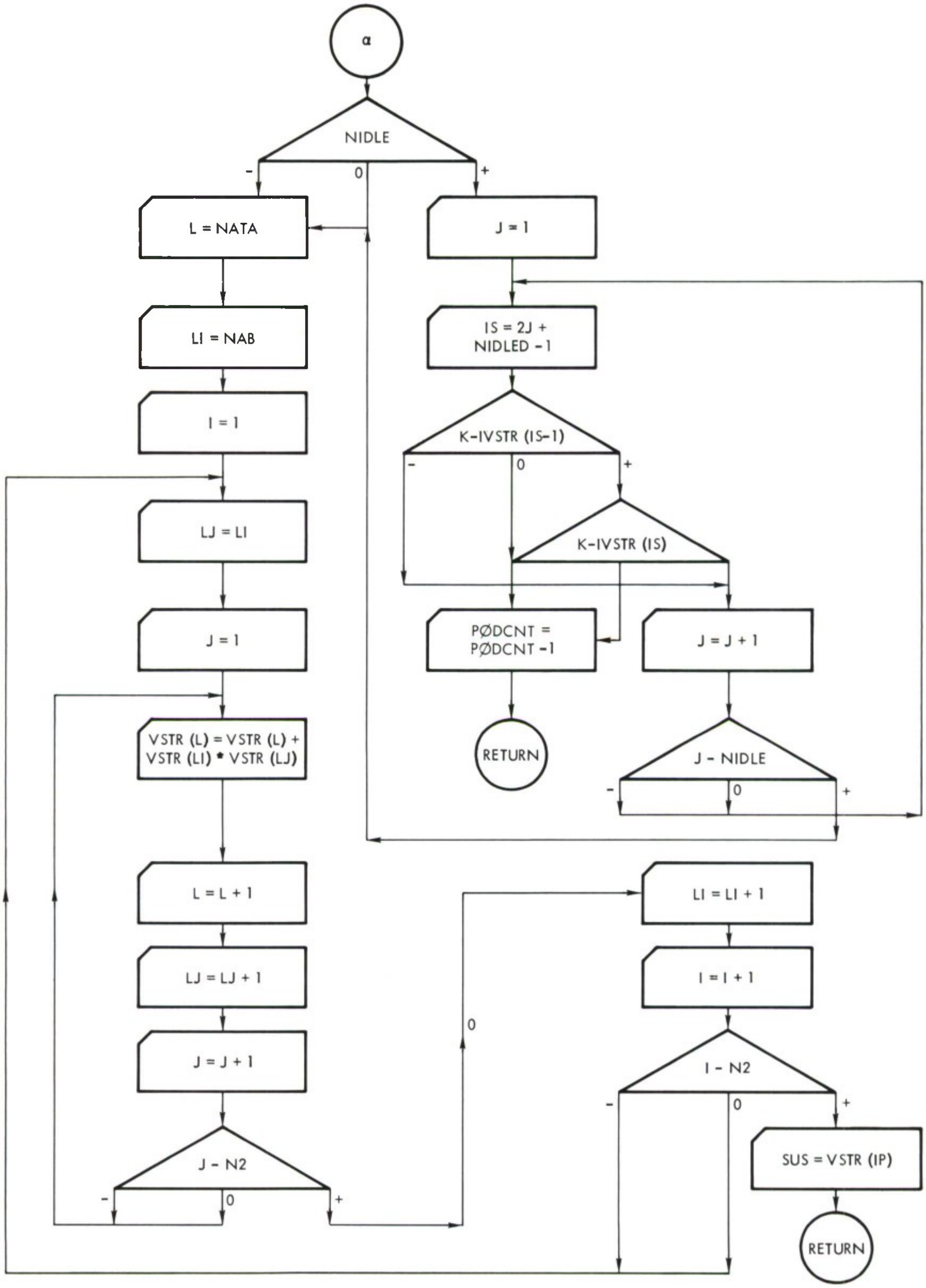


Figure 4-6 b. LEGS1 Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title

LEGS2

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

FIT (ESPØDDC)
UPDATE (ESPØDEPH)

FUNCTIONS

a) To solve an overdetermined linear system of equations $Ax = b$

b) To compute the inverse of $A^T A$

c) After solving for x, to compute $||Ax - b||^2$

USAGE

A. Calling sequence

Call LEGS2 (NDPAR, Z, SUSP, I1, I2, I4)

B. Input

1. CØMMØN

IVSTR	Fixed point variable storage
NATA	Identifies the starting location of where the upper triangular $A^T A$ is stored
NBDNS	Identifies the starting location for the bounds used by LEGS2
NPR	Number of all parameters to solve for
NR	Identifies the starting location of where the inverse $A^T A$ (in triangular form) is stored
XBSQ	Scale factor for BNDS to cause subsequent solutions to be affected by bounds

2. Calling Sequence

NDPAR	The index for variable storage where the solution vector x is to be stored
I1, I2, I4	Option control flags

C. Output

1. CØMMØN

VSTR (NDPAR)	Start of the array containing the solution vector x
VSTR (NR)	Start of an array containing $(A^TA)^{-1}$ as a lower triangular matrix

2. Calling sequence

Z	Flag to indicate if the solution was affected by the bounds. If the flag is non-zero the solution was affected by the bounds
B	Predicted SOS for the next iteration

SUBROUTINES USED

A. Library

—

B. Programs

—

EQUATIONS

To solve for differential corrections, find x so that $\|Ax - b\|^2$ is minimum under the side condition that

$$\sum_i \left(\frac{x_i}{B_i}\right)^2 \leq 1 \qquad B_1, B_2, \cdots, = \text{bounds}$$

The side condition may be described as

$$\begin{bmatrix} B_1^{-2} & 0 \; . \; . \; . \; . & \\ 0 & B_2^{-2} & \\ . & & . \\ . & & \quad . \\ & & \quad\quad . \end{bmatrix} = B^{-2} \qquad B^{-2} \text{ is a diagonal matrix}$$

where

$$x^T B^{-2} x \leq 1$$

Bounds

Define x(z) as the solution of the linear system

$$(A^TA + zB^{-2})\ X = A^Tb$$

where B^{-1} is the diagonal matrix with the (i, i) diagonal element being B_i^{-1} if $B_i > 0$ and $B_i < 0$. If $B_i = 0$, the i^{th} row and column of the augmented normal matrix is ignored and x_i is set to zero.

a) The routine finds x = x(0). If $(B^{-2}\ x, x) \leq 1 + \epsilon_1$ the solution is obtained. Otherwise

b) Define $y(z) = \left[B^{-2}\ x(z),\ x(z)\right]$. Now $y(0) > 1 + \epsilon_1$. Compare y(h), y(1 0h), y(100h), . . . , until a value of z is found with $1 - \epsilon_2 \leq y(z) \leq 1 + \epsilon_1$, in which case x(z) is the solution or until two values of z are found with $y(z_1) > 1 + \epsilon_1$ and $y(z_2) < 1 - \epsilon_2$. The required value of z is now bracketed. Then

c) Choose a value z_3 between z_1 and z_2. If $1 - \epsilon_2 \leq y(z_3) \leq 1 + \epsilon_2$, then $y(z_3)$ is the solution. Otherwise

d) Use inverse quadratic interpolation (to zero) to obtain a new guess z_4. If $1 - \epsilon_2 \leq y(z_4) \leq 1 + \epsilon_1$, then $x(z_4)$ is the solution. Otherwise

e) Select from the set z_1, z_2, z_3, z_4 the two values of z which bracket the solution most tightly. Use these values as z_1 and z_2 and go back to 3.

The iterative process will stop if the number of solutions of the linear system reaches 20.

Linear System

Let $C = A^TA + zB^{-2}$. The routine finds a matrix S with $SCS^T = D$. S is lower triangular with (-1) on the diagonal. It is easy to find S and D for a 1 x 1 matrix C. Assume S and D have been found for a k x k matrix C. Now augment C by another row and column

$$\begin{pmatrix} C & d \\ d^T & a \end{pmatrix}$$

A vector ω and a scalar β are now desired such that

$$\begin{pmatrix} S & 0 \\ \omega^T & -1 \end{pmatrix} \begin{pmatrix} C & d \\ d^T & a \end{pmatrix} \begin{pmatrix} S^T & \omega \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} D & 0 \\ 0 & \beta \end{pmatrix}$$

The requirements are satisfied by

$$\omega = S^T D^{-1} Sd$$

$$\beta = a - \omega^T d$$

The routine builds the matrix S by the above process with $k = 2, 3, \cdots, N$. The final result is a decomposition of the augmented matrix

$$\begin{pmatrix} S & 0 \\ \omega^T & -1 \end{pmatrix} \begin{pmatrix} A^T A + z\underset{\bullet}{B}^{-2} & A^T b \\ b^T A & b^T b \end{pmatrix} \begin{pmatrix} S^T & \omega \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} D & 0 \\ 0 & a \end{pmatrix}$$

and the N-dimensional vector ω which appears above is the solution vector.

<u>Predicted RMS for Next Iteration</u>

Given $b^T b$, $A^T A$, $A^T b$, X, n = total number of observations

$$\text{Predicted RMS} = \frac{1}{\sqrt{n}} \sqrt{b^T b - 2\, x^T (A^T b) + x^T (A^T A x)}$$

SUBROUTINE IDENTIFICATION

A. Title

LINES

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

The function of this subroutine is to eject a page and to print a heading top of the page after 57 lines have been printed.

USAGE

A. Calling sequence

Call LINES (A, I)

B. Input

1. CØMMØN

—

2. Calling sequence

A Line counter

I Not used

C. Output

1. CØMMØN

—

2. Calling sequence

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

PANT

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

LINES

B. Segment

ESPØDDC

C. Called by subroutines

PARSET

PUPB

RADAR

UBRERR

FUNCTION

The function is to count the number of lines and when the page is full, the Ith message is printed at the top of the next page.

USAGE

A. Calling sequence

Call LINES (A, I)

B. Input

1. CØMMØN

DCFLG	DC package control flags
KØUT	Output tape number

2. Calling sequence

A Number of lines

I Page heading number

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

LØDØBS

B. Segment

ESPØD

C. Called by subroutine

DPRØS

FUNCTION

The function is to control the logic flow in loading, storing, sorting, and printing the observations to be used in the differential correction.

USAGE

A. Calling sequence

Call LØDØBS

B. Input

1. CØMMØN

CDEG	Degrees/radian
CKMER	km/e.r.
CØMLST	Dimension of CØMMØN
CØUNT	Lines counter
DBASE	Number of days from 1950.0 to day of epoch
DBUFS	Auxiliary buffer storage
DNREV	Control cells for seven-card input
DSDAY	Epoch day, days from beginning of year
NMBER	Number of observations
NØEPØC	Flag to indicate epoch not established
PREFLG	ESPØD control flags
TEMP	Temporary storage
TEPØCH	Epoch time, minutes from midnight

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/ action messages

1. Off-line comment

"ØBS OVERFLOW CORE, NOT IN REVERSE SORT, ERROR"

2. On-line comment

"ØBS OVERFLOW CORE, NOT IN REVERSE SORT, ERROR"

3. Action

Subroutine error

SUBROUTINES USED

A. Library

GLØP
PANT

B. Program

BCDØBS	Reads one observation card
CKRSRT	Checks to see if observations are in reverse time sort
CLTIME	Converts input time into Gregorian representation
ERRØR	General error routine
FLEX	Flexowriter print routine
IPRNT	Prints header page
LINES	Keeps track of the number of lines/page
MNELTC	Converts SPADATS mean elements to Cartesian
ØBSIN	Moves observations from buffer to permanent storage
ØBSLØD	Loads observations from the SRADU tape
ØBSSRT	Sorts observations to time sequence
REWT	Rewinds observation tape
SWTSN	Monitors set up of observation data
WEØFT	Writes sentinel block on observation tape
WRTØBS	Generates observation tape

SUBROUTINE IDENTIFICATION

A. Title

LØDSEN

B. Segment

ESPØD

C. Called by subroutine

DPRØS

FUNCTION

The function is to clear out sensor and observation storage and control the logic flow in loading, converting, and compacting sensor data.

USAGE

A. Calling sequence

Call LØDSEN

B. Input

1. CØMMØN

CØMLST	Dimension of CØMMØN
DBUFS	Auxiliary buffer storage
NSSTB	Identifies the starting location where station information concerning sigmas and means of residuals are stored
NSTAT	Starting location of the master sensor table
NUBS	Starting location of the observation table
PREFLG	ESPØD control flags
TEMP	Temporary storage
VSTR	Floating point variable storage

2. Calling Sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

1. Off-line comment:

"SENSOR DATA OVERFLOWS CØMMØN, ERRØR." is printed if the sensor data overflows CØMMØN and the ERRØR subroutine is called.

2. Action:

Go to ERRØR subroutine.

SUBROUTINES USED

A. Library

GLØP

PANT

B. Program

ALSØRT	Alphanumeric sort routine
ERRØR	Error routine
SENIN	Moves sensor data from buffer to permanent storage
SENRD	Reads one sensor card from input tape
SENSCH	Searches the sensor table for a match with sensor card I. D.
SNSGET	Reads sensor information from SEAI tape

SUBROUTINE IDENTIFICATION

A. Title

MABAT

B. Segment

ESPØDEPH

C. Called by subroutine

UPDATE

FUNCTION

The function is to compute $R^* = URU^T$, where U is an N x N full matrix and R is an N x N lower triangular matrix. The result, R*, will be a lower triangular matrix

USAGE

A. Calling sequence

Call MABAT (U, I1, R, I2, RS, I3, I4)

B. Input

1. CØMMØN

2. Calling sequence

U(I1)	Starting location of U matrix
R(I2)	Starting location of R matrix
I4	Dimension of U, R and R* matrices

C. Output

1. CØMMØN

2. Calling sequence

RS (I3) Starting location of the R* matrix

D. Error/action messages

—

SUBROUTINES USED

A. Library

B. Program

SUBROUTINE IDENTIFICATION

A. Title

MAGN

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

BØDY
DAUX
DØT
RPRESS
PARØUT

FUNCTION

Function is to compute magnitude and magnitude squared of a given vector.

USAGE

A. Calling sequence

Call MAGN (A, I, B, C)

B. Input

1. CØMMØN

—

2. Calling sequence

A	Name of array containing the vector
I	Subscript locating x component of desired vector in A

C. Output

1. CØMMØN

—

2. Calling sequence

B	Magnitude of vector
C	Magnitude squared

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRT

B. Program

—

EQUATIONS

$C = R^2 = x^2 + y^2 + z^2$

$B = R = \sqrt{R^2}$

SUBROUTINE IDENTIFICATION

A. Title

MATPCH

B. Segment

ESPØDDC

C. Called by subroutines

NPRPCH

FUNCTION

The function is to punch A^TA inverse matrix in human units from VSTR (NR), if DCFLG (3) is not equal to zero, and to punch the A^TA matrix in human units from VSTR (NATA), if DCFLG (4) is not equal to zero.

USAGE

A. Calling sequence

Call MATPCH

B. Input

1. CØMMØN

DCFLG	DC package control flags
NATA	Identifies the starting location of where the upper triangular A^TA is stored
NPR	Number of all parameters to solve for
NR	Identifies the starting location of where the inverse A^TA is stored
NRTMP	Identifies starting location of temporary storage for special handling of the R matrix
NSCALE	Identifies the starting location of the list of conversion factors which convert all solution vectors and associated matrices from machine units to output units
VSTR	Floating point variable storage
KØUT	Output tape number

2. Input

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

B. Program

HUMAH	Converts vector or matrix from machine units to human units or vice versa
MØVMAT	To move a triangular matrix from A(I) to B(J)

SUBROUTINE IDENTIFICATION

A. Title

MATPT

B. Segment

1. ESPØDDC
2. ESPØDEPH

C. Called by subroutine

1. APPLY (ESPØDDC)
2. UPDATE (ESPØDEPH)

FUNCTION

The function is to print a lower triangular matrix of dimension N2 with the first element at A (N1).

USAGE

A. Calling sequence

Call MATPT (A, N1, N2)

B. Input

1. CØMMØN

TEMP	Temporary storage
IØUT	Output tape number

2. Calling sequence

A	Lower triangular matrix
N1	First element stored at A (N1)
N2	Dimension of a matrix

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library
   GLØP

B. Program
   —

SUBROUTINE IDENTIFICATION

A. Title

MLTUT

B. Segment

ESPØDEPH

C. Called by subroutine

UPDATE

FUNCTION

Function is to convert lower triangular matrix to an upper triangular matrix.

USAGE

A. Calling sequence

Call MLTUT (A, IS, B, JS, N)

B. Input

1. CØMMØN

—

2. Calling sequence

A (IS)	Starting location of A matrix (lower triangular)
N	Dimension of A and B matrices

C. Output

1. CØMMØN

2. Calling sequence

B (JS)	Starting location of upper triangular matrix

D. Error/ action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

MNELTC

B. Segment

ESPØD

C. Called by subroutine

DPRLM
LØDØBS

FUNCTION

The function is to convert the elements taken from the "SPADATS 7 CARD ELEMENT SETS" to Cartesian coordinates $(x, y, z, \dot{x}, \dot{y}, \dot{z})$.

USAGE

A. Calling sequence

Call MNELTC

B. Input

1. CØMMØN

CDEG	deg/rad
CJ	$J_2, J_3, J_4 \ldots\ldots J_{12}$
CJD50	Julian date 0^{hr} January 1, 1950
CKMER	km/e. r.
CMU	GM Earth (e. r. ^{3}/min^2)
CPI	π
C2PI	2π
DBASE	Number of days from 1950 to day of epoch
DCFLG	ESPØDDC package control flags
DNREV(1)	Type of input flag

DNREV(1) = 1. (T is input, flag)
DNREV(1) = 2. (ΔN is input, flag)
DNREV(1) = 3. (N is input, flag)

DNREV(2) Input, number

DNREV(2) = T input
DNREV(2) = ΔN input
DNREV(2) = N input

DNREV(3) Epoch revolution number

DNREV(4) Prediction flag

DNREV(4) = 1. (osculating elements)
DNREV(4) = 2. (SEAI tape)
DNREV(4) = 3. (K - 25 mean elements)
DNREV(4) = 4. (Mean elements)

DSDAY Epoch day, days from beginning of year. If DNREV(4) = 2., DSDAY contains the epoch time (days and fraction of days from 1950)

DSFDAY Epoch time, fraction of days

DYEAR Epoch year (2 digits)

KØUT Output tape number

TJDATE Julian date of midnight, epoch day

TMNEL Initial seven-card element set

If DNREV(4) = 1.

TMNEL(1) = N_o
(2) = a_o
(3) = e_o
(4) = i_o
(5) = Ω_o
(6) = ω_o
(7) = L_o

If DNREV(4) = 2.

TMNEL(1) = N_o
(2) = a_{xN}
(3) = a_{yN}
(4) = h_x
(5) = h_y
(6) = h_z
(7) = L_o
(8) = c_o

If DNREV(4) = 3. or 4.

TMNEL(1) = N_o
(2) = a_o
(3) = e_o
(4) = i_o
(5) = Ω_o
(6) = ω_o
(7) = L_o
(8) = c_o
(9) = P_n
(10) = C_n

2. Calling sequence

—

C. Output

1. CØMMØN

DNREV(5) New epoch revolution number

DNREV(6) Element set number

TMNEL(1) = N_o
(2) = a_o
(3) = e_o
(4) = i_o
(5) = Ω_o
(6) = ω_o
(7) = L_o
(8) = c_o
(9) = P_n
(10) = C_n

TNØMP(1) = α
(2) = δ
(3) = β
(4) = A
(5) = R
(6) = v

TNØMX(1) = x
(2) = y
(3) = z
(4) = $\dot{x}$
(5) = $\dot{y}$
(6) = $\dot{z}$

2. Calling sequence

—

D. Error/action messages

1. Off-line comment:
"NO CONVERGENCE IN MNELTC SUBROUTINE"

2. Action:
SUBROUTINE EXIT

SUBROUTINES USED

A. Library

CØSF
GLØP
PANT
SINF
SQRTF

B. Progtam

ASIN	Arc sine
ATNQF	Arc tangent
CTØP	Cartesian to polar coordinates
EXIT	Final exit from program
PIMØD	Takes principal value of angle between 0 and 2π
TINIT	Sets up initial time
TMSEP	Modulates initial times and sets up permanent storage

EQUATIONS

A. Preliminary (1) (If DNREV (4) = 1.)

1. Input

7-card SPWDC osculating elements

N_o	Revolution number
e_o	Eccentricity
i_o	Inclination, deg
T_o	Epoch time, day + day fraction
L_o	Mean longitude, deg
Ω_o	R.A. ascending node, deg
ω_o	Argument of perigee, deg
a_o	Semimajor axis, Earth radii
T_{oy}	Year of epoch

2. Compute

$$a_{xN} = e_o \cos \omega_o$$
$$a_{yN} = e_o \sin \omega_o$$
$$N = N_o$$
$$i_o, \ \Omega_o, \ \omega_o, \ L_o$$
$$L = L_o$$
$$\Omega = \Omega_o$$
$$i = i_o$$
$$\omega = \omega_o$$
$$e = e_o$$
$$a = a_o$$

Go To Predict Equation No. 15

B. Preliminary (2) (If DNREV (4) = 2.)

1. Input

DNREV(1), DNREV(2), DNREV(3)
From element record in memory:

a_{xN}	$\underline{N}$ component of $\underline{a}$
a_{yN}	$\underline{M}$ component of $\underline{a}$
h_x, h_y, h_z	Components of angular momentum per unit mass
L_o	Mean longitude, rad
T_o	Epoch time, days from 1950.0

c_o Rate of change of period, days/rev^2
T_{oy} Year of T_o, 1 BCD character
N_o Epoch revolution number

2. Compute

$$T_y = 365\left[(\quad)\, T_{oy} - 50\right] + \text{Integer part of } [n]$$

$$[n] = \frac{(\quad)\, T_{oy} - 48}{4} \qquad \text{If } [n] \text{ is an integer, } [n] = [n] - 1$$

$$T_o = T_o - T_y \qquad \text{(day + day fraction)}$$

$$p_o = h_x^2 + h_y^2 + h_z^2$$

$$e_o = \left(a_{xN}^2 + a_{yN}^2\right)^{1/2}$$

$$a_o = \frac{p_o}{1 - e_o^2}$$

$$q_o = a_o(1 - e_o)$$

$$i_o = \cos^{-1} \frac{h_z}{\sqrt{p_o}}$$

$$n_o = \frac{\sqrt{\mu}}{a_o^{3/2}}\left[1 - \frac{3}{4} J_2 \frac{\sqrt{1 - e_o^2}\left(1 - \frac{3}{2}\sin^2 i_o\right)}{p_o^2}\right]$$

$$\Omega_o = \tan^{-1}\left[\frac{h_x}{-h_y}\right]$$

$$C'' = \frac{-360\, n_o^2\, c_o}{\pi^2}$$

$$P_N = \frac{2\pi}{n_o}\left[1 - \frac{3}{4} J_2\left(\frac{1}{p_o^2}\right)(4 - 5\sin^2 i)\right]$$

$$C_N = c_o$$

$$\omega_o = \tan^{-1}\left(\frac{a_{yN}}{a_{xN}}\right) \quad \text{(rad)}$$

3. Flag

DNREV (4) = 2.

Go to Predict Equation No. 3

C. Preliminary (3) (If DNREV (4) = 3.)

1. Input

DNREV(1), DNREV(2), DNREV(3)
7-card element set (mean K-25)
N_o Revolution number
e_o Eccentricity
i_o Inclination, deg
T_o Epoch time, day + day fraction
L_o Mean longitude, deg
Ω_o R. A. ascending node, deg
ω_o Argument of perigee, deg
c_o Rate of change period, day/rev^2
a_o (K - 25) semimajor axis, earth radii
P_n Nodal period, days/rev
C_N Rate of change, nodal period, days/rev^2
T_{oy} Year of epoch

2. Compute

$p_o = a_o\left(1 - e_o^2\right)$
$q_o = a_o(1 - e_o)$
$i_o, \Omega_o, \omega_o, L_o \longrightarrow$ Radians

3. Flag

(If DNREV (4) = 3.)
Go to Predict Equation No. 1

D. Preliminary(4) (If DNREV (4) = 4.)

1. Input

DNREV(1), DNREV(2), DNREV(3)
7-card element set (mean)
N_o Revolution number
e_o Eccentricity
i_o Inclination, deg
T_o Epoch time, day + day fraction
L_o Mean longitude, deg
Ω_o R. A. ascending node, deg
ω_o Argument of perigee, deg

c_o	Rate of change of period, days/rev^2
a_o	Semimajor axis, Earth radii
P_N	Nodal period, days/rev
C_N	Rate of change nodal period, days/rev^2
T_{oy}	Year of epoch

2. Compute

$$p_o = a_o\left(1 - e_o^2\right)$$

$$a_o = a_o\left[1 - \frac{3}{2}\frac{J_2}{p_o^2}\left(1 - \frac{3}{2}\sin^2 i_o\right)\sqrt{1 - e_o^2}\right]$$

$$q_o = a_o(1 - e_o)$$

$i_o,\ L_o,\ \Omega_o,\ \omega_o$

Go to Predict Equation No. 1

3. Flag

DNREV (4) = 3.

E. Predict

1) $$n_o = \frac{\sqrt{\mu}}{a_o^{3/2}}\left[1 - \frac{3}{4}\frac{J_2}{P_o^2}\left(1 - \frac{3}{2}\sin^2 i_o\right)\sqrt{1 - e_o^2}\right]$$

2) $$C'' = \frac{-360\ n_o^2\ C_o}{\pi^2}$$

3) $A = 8$

$n_d = 0.072220521$

$$d = A(C'')^2\left[1 + \frac{n_o}{3(n_d - n_o)}\right]$$

4) $$a = a_o\left[1 + 2C''\ \Delta T\ (1440) + 3d\ (\Delta T)^2\ (1440)^2\right]^{-2/3}$$

5) $$e = 1 - \frac{q_o}{a} \qquad a > q_o$$

$$e = o \qquad a \leq q_o$$

6) $\dot{\Omega} = -\frac{3}{2} \frac{J_2}{p_o^2} n_o \cos i_o \,(1440)$

7) $\dot{\omega} = \frac{3}{4} \frac{J_2}{p_o^2} n_o \,(4 - 5 \sin^2 i_o)\,(1440)$

8) $\Omega = \Omega_o + \dot{\Omega}\Delta T$

9) $\Delta\omega = \dot{\omega}\Delta T$

10) $a_{xN} = e \cos \omega_o \cos \Delta\omega - e \sin \omega_o \sin \Delta\omega$

11) $a_{yN} = e \cos \omega_o \sin \Delta\omega + e \sin \omega_o \cos \Delta\omega$

$$- \frac{1}{2} \frac{J_3}{J_2} \frac{\sin i_o}{p_o}$$

12) $\Delta\pi = \frac{3}{2} \frac{J_2}{p_o^2} \left(2 - \frac{5}{2} \sin^2 i_o - |\cos i_o|\right)$

13) $L_3 = -\frac{1}{4} \frac{J_3}{J_2} \frac{a_{xN} \sin i_o}{p_o} \left[\frac{3 + 5\,|\cos i_o|}{1 + |\cos i_o|}\right]$

14) $L = L_o + L_3 + (n_o)(\Delta T)(1440)$

$$\left[(1 + \Delta\pi) + (C'')(\Delta T)(1440) + (d)(\Delta T)^2\,(1440)^2\right]$$

15) $U = L - \Omega \qquad i \leq 90^o$

$U = L + \Omega \qquad i > 90^o$

16) $(E + \omega)' = v + a_{xN} \sin (E + \omega)_n - a_{yN} \cos (E + \omega)_n$

first approximation

$(E + \omega)_1 = U$

17) $(E + \omega)_{n+1} = \dfrac{\left[a_{xN}\,(E + \omega)_n + a_{yN}\right] \cos (E + \omega)_n + \left[a_{yN}\,(E + \omega)_n - a_{xN}\right] \sin (E + \omega)_n - U}{a_{xN} \cos (E + \omega)_n + a_{yN} \sin (E + \omega)_n - 1}$

continue until $|(E + \omega)' - (E + \omega)_n| < 5.E - 7$

18) $e \sin E = (E + \omega)_n - v$

19) $e \cos E = a_{XN} \cos (E + \omega) + a_{YN} \sin (E + \omega)$

20) $r = a(1 - e \cos E)$

21) $\dot{r} = \frac{\sqrt{\mu a}}{r} \; e \sin E$

22) $r\dot{v} = \frac{\sqrt{\mu a}}{r} \sqrt{1 - e^2}$

23) $\cos u = \frac{a}{r} \left[\cos (E + \omega) - a_{xN} + a_{xN} \left(\frac{e \sin E}{1 + \sqrt{1 - e^2}} \right) \right]$

24) $\sin u = \frac{a}{r} \left[\sin (E + \omega) - a_{yN} - a_{yN} \left(\frac{e \sin E}{1 + \sqrt{1 - e^2}} \right) \right]$

25) $\underline{N} = \begin{cases} N_x = \cos \Omega \\ N_y = \sin \Omega \\ N_z = 0 \end{cases}$

26) $M = \begin{cases} M_x = -\sin \Omega \cos i \\ M_y = \cos \Omega \cos i \\ M_z = \sin i \end{cases}$

27) $\underline{U} = \underline{N} \cos u + \underline{M} \sin u$

28) $\underline{V} = -\underline{N} \sin u + \underline{M} \cos u$

29) $\underline{r} = r\underline{U}$

30) $\dot{\underline{r}} = \dot{r}\underline{U} + r\dot{v}\underline{V}$

F. Revolution Number (If DNREV (4) = 2.)

1) $u = \tan^{-1} \left[\frac{\sin u}{\cos u} \right] \qquad 0 \leq u \leq 2\pi$

2) $\Delta N_1 = \left[(L - L_o - \dot{\Omega} \Delta T - \Delta\omega) \bmod 2\pi \right] + \left[(\omega_o + \Delta\omega) \bmod 2\pi \right]$

ΔN_1 is the number of times these arguments are reduced by 2π (to reach $0 \leq \theta \leq 2\pi$)

3) $u \geq \omega$

If yes, $\Delta N_1 = \Delta N_1$

If no, $\Delta N_1 = \Delta N_1 + 1$

4) $N = N_o + \Delta N_1$

SUBROUTINE IDENTIFICATION

A. Title

MØVE

B. Segment

ESPØD

C. Called by subroutine

ASSIGN

FUNCTION

This subroutine moves storage between A(I) and A(J) forward or backward N cells depending on the flag M.

USAGE

A. Calling sequence

Call MØVE (A, I, J, N, M)

B. Input

1. CØMMØN

—

2. Calling sequence

A	Block of storage
I	Identifies the old starting location of the "A" block
J	Identifies the new starting location of the "A" block
N	Number of cells to be moved
M	If M = 1, block should be moved forward N cells; if M = -1, block should be moved backward N cells

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

MØVE

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

MØVEVS

B. Segment

ESPØDDC

C. Called by subroutines

UBSGET

FUNCTION

This subroutine moves observation set from variable to working storage, sets up observational sigmas, and sets up G_s for gross outlier rejection criterion.

USAGE

A. Calling sequence

Call MØVEVS (J)

B. Input

1. CØMMØN

DBASE	Number of days from January 1, 1950 to day of epoch
VSTR (NUBS)	Array containing observation sets (see format of observations on the following pages)

2. Calling sequence

J	Index for the next observation set to be picked up out of array VSTR (NUBS)

C. Output

1. CØMMØN

PUBS(1)	Sensor number
(2)	Observation time, min from 0^h day of epoch
(3)	Range, e.r.
(4)	Azimuth, rad
(5)	Elevation, rad
(6)	Range rate, e.r./min
(7)	Hour angle, rad, if applicable
(8)	Declination, rad, if applicable

PSIG(1)	σ_R, e.r.	observation weights
(2)	σ_A, rad	
(3)	σ_E, rad	
(4)	σ_{RDT}, e.r./min	
(5)	σ_{HA}, rad	
(6)	σ_{DEC}, rad	
PKSUBS	G_s (gross outlier rejection criterion)	

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

Observation set as recorded on LØG 7 and as maintained in the VSTR (NUBS) array

Word									
1	G_s —binary integer at B23			O	N	N	N		G_s; sensor number (BCD)
2	CF	CL	ØT	E	T	S	S	S	(see index 1)
3	48 bit floating point number								Time, days and fractions of days from Jan. 1, 1950
4	R	Ø	Ø	Ø	Ø	Ø	E	A	(see index 2)
5	48-bit floating point number								Elevation, declination
6	48-bit floating point number								Azimuth, hour angle
7	48-bit floating point number								Slant range
8	48-bit floating point number								Range rate
9	σ_R				σ_A				*Observation weights assigned at observation processing time.
10	σ_E				$\sigma_{\dot{R}}$				

INDEX 1

CF Maximum frequency shift indicator

CL Classification Δ = unclassified
1 = classified

ØT Observation time

0 range rate only
1 azimuth and elevation
2 azimuth, elevation, range
3 azimuth, elevation, range, range rate
5 right ascension and declination

ET Equipment type

A Accuracy

INDEX 2

R Association indicator

E Equinox

A Accuracy

*These weights are stored as binary integers, two per word (one at a B23 and the other at a B47.) The weights are these integers converted to floating point numbers and then divided by 10^4. For optical data the first word contains weights for field reduced RA and DEC and the second word contains weights for precision reduced RA and DEC.

SUBROUTINE IDENTIFICATION

A. Title

MØVMAT

B. Segment

ESPØDDC

C. Called by subroutine

APPLY
MATPCH

FUNCTION

The function is to move a triangular matrix of dimension K from A(I) to B(J).

USAGE

A. Calling sequence

Call MØVMAT (A, I, B, J, K)

B. Input

1. CØMMØN

—

2. Calling sequence

A	Triangular matrix
I	Starting location of matrix to be moved
K	Dimension of A matrix
J	Starting location where matrix is to be stored

C. Output

1. CØMMØN

—

2. Calling sequence

B   Relocated matrix

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

MULT

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

RØTRU

FUNCTION

Function is to multiply a given 3 x 3 matrix times a succession of 1 x 3 vectors.

USAGE

A. Calling sequence

Call MULT (S, A, B, I, NCØL)

B. Input

1. CØMMØN

—

2. Calling sequence

S	Address of the 3 x 3 matrix stored by rows
A	Address of a succession of column vectors
I	Location of the x component of the first vector of A to be used
NCØL	The number of successive column vectors of A to be used

C. Output

1. CØMMØN

—

2. Calling sequence

B	Address of the array containing the resultant product, stored by rows

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

None

SUBROUTINE IDENTIFICATION

A. Title

NPRPCH

B. Segment

ESPØDDC

C. Called by subroutine

FIT

FUNCTION

Function is to punch the values of the solution parameters to be used on the next iteration, the associated bounds, the A^TA matrix and the A^TA inverse matrix if required, in human units.

USAGE

A. Calling sequence

Call NPRPCH

B. Input

1. CØMMØN

CDAD2M	$C_DA/2M$
CDVAR	ϵ = drag variation
DHEAD	Header from JDC card
DVEHN	Vehicle number and name (BCD)
IVSTR	Fixed point variable storage
NBDNS	Identifies the starting location for the bounds
NDPR	Number of all differential + initial parameters to solve for (Category 1)
NICPR	Number of initial condition parameters to solve for
NIDP	Identifier for table indicating Category 1 type variables to be solved for
NITCT	Iteration counter
NPAR	Identifies the starting location for the parameter list

NPR	Number of all parameters to be solved for
NRTMP	Starting location of temporary storage for special handling of the R matrix
NSCALE	Starting location of the list of conversion factors
TNØMP	Initial polar coordinates
VSTR	Floating point variable storage
KØUT	Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

B. Program

1. HUMAH — Converts vector or matrix from machine units to human units or vice versa

2. MATPCH — To punch A^TA and $(A^TA)^{-1}$ at the end of each iteration

SUBROUTINE IDENTIFICATION

A. Title

ØBSIN

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

Function is to move data from temporary storage (TEMP) to permanent storage (Z). This subroutine also converts temporary storage to internal units.

USAGE

A. Calling sequence

Call ØBSIN (Z, ISTART)

B. Input

1. CØMMØN

CDEG	Degrees/radian	
CJD50	Julian date January 0, 1950	
CKMER	km/e.r.	
TEMP(30)	Satellite number	
(31)	Equipment type	
(32)	Station number	
(33)	Year	
(34)	Month	
(35)	Day	
(36)	Hour	
(37)	Minutes	
(38)	Seconds	
(39)	E or δ (degrees)	
(40)	A or α (degrees)	
(41)	R, slant range (km)	
(42)	$\dot{R}$ (km/sec)	
(43)	Code for field 10	
(44)	At observation time	brightness
(45)	Maximum	brightness
(46)	Minimum	brightness
(47)	Time interval	
(48)	Date or line number	
(49)	Message number	
(50)	Equinox	
(51)	Year	
(52)	Observation number	
(53)	Card type	

2. Calling sequence

ISTART	Starting location of Z

C. Output

1. CØMMØN

—

2. Calling sequence

Z(ISTART)	Station number
Z(ISTART+1)	Satellite number
Z(ISTART+2)	Time (days since 1950)
Z(ISTART+3)	Card type
Z(ISTART+4)	E or δ (radians)
Z(ISTART+5)	A or α (radians)
Z(ISTART+6)	R, slant range (e.r.)
Z(ISTART+7)	$\dot{R}$, range rate (e.r./min)
Z(ISTART+8)	Brightness
Z(ISTART+9)	Observation type

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

TIME	Compute Julian date

SUBROUTINE IDENTIFICATION

A. Title

ØBSLØD

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

This subroutine loads observations from the SRADU tape into core, for the satellite number found in DVEHN. When core is filled the subroutine exits. Multiple entrances into this subroutine are permitted to load all the data from tape.

USAGE

A. Calling sequence

Call ØBSLØD (SEØF)

B. Input

1. CØMMØN

CØMLST	Dimension of CØMMØN
DVEHN	Vehicle number and name (BCD)
NMBER	Number of observations
VSTR	Floating point variable storage

2. Calling sequence

SEØF	Sentinel block detection flag

C. Output

1. CØMMØN

TEMP(100)	Starting location of the observations from tape

2. Calling sequence

—

D. Error/action messages

1. Off-Line Comment:

"ERROR. NO OBS ON SRADU FOR SATELLITE NO. ______"

2. On-Line Comment:

"ERROR. NO OBS ON SRADU FOR SATELLITE NO. ______"
"TYPE GO TO REREAD TAPE, STOP FOR NEXT CASE"

3. Action:

Subroutine error

SUBROUTINES USED

A. Library

GLØP
READT
STARTRD
STØPGØ
TAPCK
ZCHEK

B. Program

ERRØR	Error routine
FLEX	Flexowriter print routine
IDSUB	Strip blanks from I. D.

SUBROUTINE IDENTIFICATION

A. Title

ØBSSRT

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

Function is to sort the observations timewise with respect to the number of days from 1950.0 to the day of epoch.

USAGE

A. Calling sequence

Call ØBSSRT (A, ISTART, IFINAL)

B. Input

1. CØMMØN

DBASE	Number of days from 1950.0 to day of epoch
DHOUR	Epoch hour
DMIN	Epoch minute
DSEC	Epoch second
TEMP	Temporary storage

2. Calling sequence

A	Storage array
ISTART	Identifier for starting location of array in A storage
IFINAL	Identifier for ending location of array in A storage

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

ØBSSRT

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

ØUTER

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

BØDY
DRAG
VAREQ

FUNCTION

Function is to compute the "outer product," i.e., the 3 x 3 matrix product, which results when a 3 x 1 column vector is multiplied times a 1 x 3 row vector.

USAGE

A. Calling sequence

Call ØUTER (A, I, B, J, C)

B. Input

1. CØMMØN

—

2. Calling sequence

A	Address of the 3 x 1 column vector array
I	Location of first element in A
B	Address of 1 x 3 row vector array
J	Location of first element in B

C. Output

1. CØMMØN

—

2. Calling sequence

C	Address of 3 x 3 array to which the outer product is added

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

None

SUBROUTINE IDENTIFICATION

A. Title

ØUTPT

B. Segment

ESPØDEPH

C. Called by subroutine

TCØMP

FUNCTION

Function is to punch on DS-2 the sets of x_T, y_T, z_T, t_D, t_{Df} generated by TCØMP. These punched cards may be used as inputs to the GIPAR program.

USAGE

A. Calling sequence

Call ØUTPT

B. Input

1. CØMMØN

TRAJX	(1)	x_T (e. r.)
	(2)	y_T (e. r.)
	(3)	z_T (e. r.)
	(4)	t_D (days)
	(5)	t_{FD} (fraction of days)
SEQ		Sequence number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINE IDENTIFICATION

A. Title

PARØUT

B. Segment

ESPØDDC

C. Called by subroutine

RADR

FUNCTION

Function is to compute the following items for the residuals print.

1. Residuals in U, V, W system or on option residuals in the S, T, W system. A second option is provided to show how much the sensor location, in terms of ϕ, λ, and h, would have to be moved in order to make residual errors equal to zero
2. The vector magnitude of the residuals in the U, V, W system
3. The in-plane time differential between the measured and computed positions
4. The argument of latitude of the computed position
5. The out-of-plane angle beta.

USAGE

A. Calling sequence

Call PARØUT

B. Input

1. CØMMØN

DCFLG(5)	= 0 Print U, V, W residuals = 1 Print S, T, W residuals = 2 Print ϕ, λ, h residuals
PCMR	R, computed slant range
PRESD	Array containing observation residuals (ΔR, ΔA or ΔHA, ΔE or ΔDEC $\Delta\dot{R}$)
PSTAT	Working storage of sensor information

PUBS	Array containing sensor number, time, R, A, E, $\dot{R}$, HA, DEC. All observations are measured.
PWI	Vector (w_1, w_2, w_3)
CMU	GM of the Earth (e.r.^{3}/min^2)
SNALF	$\sin \alpha$ where $\alpha = \alpha_{go} + \lambda_s + w_e t$
CSALF	$\cos \alpha$
TRAJX	Array containing $x, y, z, \dot{x}, \dot{y}, \dot{z}$ at time t_ϕ (observation time)

2. Calling sequence

—

C. Output

1. CØMMØN

PREDT(1)	ΔR = slant range residual (km)
(2)	ΔA or ΔHA = azimuth or hour angle residual (deg)
(3)	ΔE or ΔDEC = elevation or declination residual (deg)
(4)	$\Delta \dot{R}$ = range rate residual (km/sec)
(5)	Δu, $\Delta \mathbf{s}$, or $\Delta \phi$ (km, km, deg)
(6)	Δv, $\Delta \mathbf{t}$, or $\Delta \lambda$ (km, km, deg)
(7)	Δw, $\Delta \mathbf{w}$, or Δh (km, km, meters)
(8)	$v \text{ mag} = \sqrt{\Delta u^2 + \Delta v^2 + \Delta w^2}$ (km)
(9)	Δt = in-plane time differential (min)
(10)	U = argument of latitude (deg)
(11)	β = out-of-plane angle (deg)

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF

SINF

SQRTF

B. Program

ASIN	Arcsin
ATNQF	Arc tangent
DØT	Dot product
PIMØD	Puts angle between 0 and 2π
MAGN	Computes $\lvert y \rvert$ and $\lvert y \rvert^2$ of a vector y (y_1, y_2, y_3)
XCRØSS	Cross product routine
YHADEC	Computes the y vector given hour angle and declination measurements
YRAE	Computes the y vector given range, azimuth, elevation measurements

EQUATIONS

$$\dot{w} = (\dot{w}_1, \dot{w}_2, \dot{w}_3)$$

where

$$\dot{w}_1 = \dot{x} \cos a + \dot{y} \sin a$$

$$\dot{w}_2 = -\dot{x} \sin a + \dot{y} \cos a$$

$$\dot{w}_3 = \dot{z}$$

Compute u, v, w (up, down, cross)

$$\overrightarrow{UP} = \frac{\vec{w}}{|\vec{w}|} = \frac{\vec{w}}{r} \qquad \text{where } r = \sqrt{w_1^2 + w_2^2 + w_3^2}$$

$$\overrightarrow{DØWN} = \frac{\vec{u}}{|\vec{u}|} \qquad \text{where } u = \dot{\vec{w}} - \eta\,\vec{w} \text{ and } \eta = \frac{\dot{\vec{w}}\cdot\vec{w}}{r^2}$$

$$\overrightarrow{CRØSS} = \overrightarrow{UP} \times \overrightarrow{DØWN}$$

If DCFLG(5) = 1, compute s, t, w

where

$$\vec{s} = \frac{\vec{u}}{|\vec{u}|} \qquad \text{where } \vec{u} = \vec{w} - \eta\,\dot{\vec{w}} \text{ and } \eta = \frac{\dot{\vec{w}}\cdot\vec{w}}{v^2}$$

$$\text{with } V = \sqrt{\dot{w}_1^2 + \dot{w}_2^2 + \dot{w}_3^2}$$

$$\vec{t} = \frac{\dot{\vec{w}}}{|\dot{\vec{w}}|} = \frac{\dot{\vec{w}}}{V}$$

$$\vec{w} = \vec{s} \times \vec{t}$$

Compute vector y from either subroutine YRAE or subroutine YHADEC then, if:

DCFLG(5) = 0, compute ΔUP, ΔDOWN, ΔCROSS

$$\Delta UP = (\vec{y} - \vec{w}) \cdot \overrightarrow{UP}$$

$$\Delta DOWN = (\vec{y} - \vec{w}) \cdot \overrightarrow{DOWN}$$

$$\Delta CROSS = (\vec{y} - \vec{w}) \cdot \overrightarrow{CROSS}$$

or if DCFLG(5) = 1, compute ΔS, ΔT, ΔW

$$\Delta S = (\vec{y} - \vec{w}) \cdot \vec{s}$$

$$\Delta T = (\vec{y} - \vec{w}) \cdot \vec{t}$$

$$\Delta W = (\vec{y} - \vec{w}) \cdot \vec{w}$$

or if DCFLG(5) = 2, compute ΔStation Latitude, ΔStation Logitude, ΔStation Altitude

$$\overrightarrow{w_s^*} = \overrightarrow{w_s} + \vec{w} - \vec{y}$$

$$R^* = |\overrightarrow{w_s^*}| \qquad R_s = |\overrightarrow{w_s}|$$

$$\phi = \tan^{-1}\left[\frac{w^*_{s3}}{w^*_{s1}}\right] \qquad \phi = 0$$

$$\theta^* = \sin^{-1}\left[\frac{w^*_{s3}}{R^*}\right] \qquad \theta = \tan^{-1}\left[\frac{w_{s3}}{w_{s1}}\right]$$

and

$$\Delta \text{ Station Latitude} = \theta^* - \theta$$

$$\Delta \text{ Station Longitude} = \phi^* - \phi$$

$$\Delta \text{ Station Altitude} = R^* - R$$

Compute vector magnitude, β, and Δu

$$VM = |\vec{y} - \vec{w}|$$

$$\beta = \sin^{-1}\left[\frac{\overrightarrow{cross} \cdot \vec{y}}{|y|}\right]$$

$$\Delta u = \tan^{-1}\left[\frac{\Delta DØWN}{\vec{y} \cdot \overrightarrow{UP}}\right]$$

Compute argument of latitude

$$u = \tan^{-1}\left[\frac{UP_3}{DØWN_3}\right]$$

Begin computations for Δ-t

$$\lambda = \frac{rv^2}{\mu}, \qquad a = \frac{r}{2 - \lambda}, \qquad n = \frac{\sqrt{\mu}}{a^{3/2}}$$

$$e = \left[(1 - \lambda)^2 + 2(2 - \lambda)\left(\frac{w \cdot \dot{w}}{rv}\right)\right]^{1/2}$$

$$\cos v_1 = \frac{a(1 - e^2) - r}{re}$$

$$\sin v_1 = \pm\sqrt{1 - \cos^2 v_1} \qquad + \text{ if } \vec{w} \cdot \dot{\vec{w}} > 0$$

$$v_2 = v_1 + \Delta u$$

for $a > 0$

$$\cos E_j = \frac{r \cos v_j + ae}{a}$$

$$\sin E_j = \frac{r \sin v_j}{a\sqrt{1 - e^2}}$$

$$E_j = \tan^{-1}\left[\frac{\sin E_j}{\cos E_j}\right]$$

$$M_j = E_j - e \sin E_j$$

for $a < 0$

$$\cosh F_j = \frac{r \cos v_j + ae}{a}$$

$$\sinh F_j = \frac{r \sin v_j}{-a\sqrt{e^2 - 1}}$$

$$F_j = \tanh^{-1}\left[\frac{\sinh F_j}{\cosh F_j}\right]$$

$$M_j = e \sinh F_j - F_j$$

finally

$$\Delta t = \frac{M_2 - M_1}{n}$$

SUBROUTINE IDENTIFICATION

A. Title

PARSET

B. Segment

ESPØDDC

C. Called by subroutine

INTEG

FUNCTION

This subroutine sets up the PSTAT array with sensor information from the master sensor table for a given sensor number. It checks to see if either latitude, longitude, altitude, or time biases are being solved for by this sensor and if so, updates the PSTAT table before returning to the main sequence.

USAGE

A. Calling sequence

Call PARSET

B. Input

1. CØMMØN

CØUNT	Lines counter
IVSTR	Fixed point variable storage
NPBIS	Identifies table for definition of Category 2 variables
NPRCD	Identifies table for definition of Category 2 variables to be solved for
NSTAT	Identifies the starting location of the master sensor table
PLSTSN	Name of the last sensor processed by RADR
PSIG	Sigma list for current station and associated time and observations

PUBS	Current observations and time table
TEMP	Temporary storage
TG	Time to integrate to
TMBIS	Current estimate of time bias for the observation time being considered
VSTR	Floating point variable storage
CAE	a_e
CBE	b_e
CDEG	Degrees/radian
CKMER	km/e.r.
CWE	Earth's rotational rate
KØUT	Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

PSTAT	(1)	ϕ_s	sensor latitude (rad)
	(2)	λ_s	sensor longitude (rad)
	(3)	h	sensor altitude (e.r.)
	(4)	$\cos \phi_s$	
	(5)	$\sin \phi_s$	
	(6)	$\alpha_{go} + \lambda_s$	
	(7)	ω_1^s	coordinates this sensor in the W system (e.r.)
	(8)	ω_3^s	
	(9)	Code word (see definition of IVSTR(NPRCD) array)	
TG		Observation time (adjusted by approximate time bias if applicable).	

2. Calling sequence

—

D Error/action messages

"SENSØR xxx NØT IN MASTER SENSOR LIST"

After this message is printed control is returned to the main sequence and the next observation time is selected.

SUBROUTINES USED

A. Library

CØSF
GLØP
SINF
SQRTF

B. Program

LINES	Ejects a page and prints a header
PIMØD	Takes principal value of angle between 0 and 2π

EQUATIONS

Where applicable

$$\phi_s = \phi_{so} + \Delta\phi_s$$

$$\lambda_s = \lambda_{so} + \Delta\lambda_s$$

$$h = h_o + \Delta h$$

$$\cos\phi = \cos(\phi_{so} + \Delta\phi_s)$$

$$\sin\phi = \sin(\phi_{so} + \Delta\phi_s)$$

$$\alpha_{go} + \lambda_s = \alpha_{go} + \lambda_o + \Delta\lambda_s$$

$$w_1^s = \left[a_e A_s + (h_o + \Delta h_s)\right] \cos(\phi_{so} + \Delta\phi_s)$$

$$w_3^s = \left[b_e B_s + (h_o + \Delta h_s)\right] \sin(\phi_{so} + \Delta\phi_s)$$

where

$$A_s = \left[\cos^2(\phi_{so} + \Delta\phi_s) + \left(\frac{b_e}{a_e}\right)^2 \sin^2(\phi_{so} + \Delta\phi_s)\right]^{-1/2}$$

$$B_s = \left[\sin^2(\phi_{so} + \Delta\phi_s) + \left(\frac{a_e}{b_e}\right)^2 \cos^2(\phi_{so} + \Delta\phi_s)\right]^{-1/2}$$

$$a_e = 1.0$$

$$b_e = a_e(1. - \epsilon)$$

The ϕ_{so}, λ_{so}, and h_o are the latitude, longitude, and altitude of the sensor taken from the master sensor list. The $\Delta\phi_s$, $\Delta\lambda_s$, and Δh_s are the current estimates of the biases in the sensor latitude, longitude, and altitude as computed by the differential correction. If these biases are not being solved for, the above equations are ignored and the corresponding entries in PSTAT are taken directly from the master sensor list.

SUBROUTINE IDENTIFICATION

A. Title

PIMØD

B. Segment

ESPØD

ESPØDDC

ESPØDEPH

C. Called by subroutines

—

FUNCTION

Function is to get the positive argument of an angle in radians between 0 and 2π.

USAGE

A. Calling sequence

PIMØD(A)

B. Input

1. CØMMØN

   C2PI 2π

2. Calling sequence

   A Angle in radians

C. Output

1. CØMMØN

   —

2. Calling sequence

   A Positive angle between 0 and 2π in radians

SUBROUTINE IDENTIFICATION

A. Title

PHEAD

B. Segment

ESPØDDC

C. Called by subroutines

RADR

FUNCTION

Function is to print the header for the residuals.

USAGE

A. Calling sequence

Call PHEAD

B. Input

1. CØMMØN

DCFLG	ESPØDDC control flags
KØUT	Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

PLTEL

B. Segment

ESPØDEPH

C. Called by subroutine

TPRNT

FUNCTION

Function is to convert polar coordinates to both classical and indeterminacy free elements.

USAGE

A. Calling sequence

Call PLTEL (A, B, I)

B. Input

1. CØMMØN

CMU	GM Earth (e. r.^{3}/min^2)
CPI	π
C2PI	2π
TEMP	Temporary storage
TG	Time to integrate to

2. Calling sequence

(1) α
(2) δ
(3) β
(4) A
(5) R
(6) v

C. Output

1. CØMMØN

—

2. Calling sequence

$$\left.\begin{array}{r} B(1) \\ B(2) \\ (3) \end{array}\right\} \underline{u}_o$$

(4) D_o
(5) r_o
(6) $1/a$
(7)
(8) $\bar{P}\ \underline{v}_o$
(9)
(10) a
(11) e
(12) i
(13) Ω
(14) ω Given only if $\frac{1}{a} < 0$
(15) u_o
(16) M_o
(17) T
I $I = 0$, parabolic orbit
$I = -1$, hyperbolic orbit
$I = +1$, ellipse

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF
SINF
SQRTF

B. Program

ATNQF	Arc tangent routine
PIMØD	Takes principle value of angle between 0 and 2π

EQUATIONS

1. $$\underline{u}_o = \begin{bmatrix} \cos\delta\cos\alpha \\ \cos\delta\sin\alpha \\ \sin\delta \end{bmatrix} = \begin{bmatrix} u_{x_o} \\ u_{y_o} \\ u_{z_o} \end{bmatrix}$$

2. $D_o = \dfrac{r\ v \cos\beta}{\sqrt{\mu}}$

3. $r_o = r$

4. $\dfrac{1}{a} = \dfrac{2}{r_o} - \dfrac{v^2}{\mu}$

5. $$\sqrt{P}\ v_o = \begin{bmatrix} \dfrac{1}{\sqrt{\mu}} r_o \left[v \sin\beta\,(-\sin A \sin\alpha - \cos A \cos\alpha) + v \cos\beta \cos\delta \cos\alpha \right] - D_o \cos\delta \cos\alpha \\ \dfrac{1}{\sqrt{\mu}} r_o \left[v \sin\beta\,(\sin A \cos\alpha - \cos A \sin\alpha) + v \cos\beta \cos\delta \sin\alpha \right] - D_o \cos\delta \sin\alpha \\ \dfrac{1}{\sqrt{\mu}} r_o \left[v \sin\alpha\,(\sin 2 + \cos A \cos\delta) \right] - D_o \sin\delta \end{bmatrix}$$

6. Test: $\frac{1}{a}$ 0, "parabolic orbit"
-, "hyperbolic orbit"
+, "ellipse" and go to 7.

7. $a = \left(\dfrac{1}{a}\right)^{-1}$

8. $$e = \left[\left(\frac{D_o}{\sqrt{a}}\right)^2 + \left(1 - \frac{r_o}{a}\right)^2 \right]^{1/2}$$

9. $i = \cos^{-1}\left(u_{x_o} v_{y_o} - u_{y_o} v_{x_o}\right)$

10. $$\Omega = \tan^{-1}\left[\frac{u_{y_o} v_{z_o} - u_{z_o} v_{y_o}}{u_{x_o} v_{z_o} - u_{z_o} v_{x_o}}\right]$$

11. $E_o = \tan^{-1}\left[\frac{D_o\sqrt{a}}{a - r}\right]$

$M_o = E_o - e \sin E_o$

12. $T = \frac{a^{3/2}}{\sqrt{\mu}} M_o - t_o$

13. $u_o = \cos^{-1}(\cos\delta\cos\alpha\cos\Omega + \cos\delta\sin\alpha\sin\Omega)$

Test: If δ is negative, $u_o = 2\pi - u_o$

If δ is positive, u_o = O'K'

14. $v_o = \cos^{-1}\left[\frac{a}{r}(\cos E - e)\right]$

a. For $0 \leq M_o \leq 180^o$

$v_o = v_o$

b. For $180^o < M_o < 360^o$

$v_o = 2\pi - v_o$

$\omega = u_o - v_o$

SUBROUTINE IDENTIFICATION

A. Title

PØLY

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

DYNAT (ESPØDDC, ESPØDEPH)

FUNCTION

Function is to evaluate a polynomial, given the coefficients, the number of coefficients and the independent variable.

USAGE

A. Calling sequence

Call PØLY (CØEFS, NCØEFS, ANS, ARG)

B. Input

1. CØMMØN

—

2. Calling sequence

CØEFS	Table of coefficients (maximum of 15)
NCØEFS	Number of coefficients
ARG	Argument (independent variable)

C. Output

1. CØMMØN

—

2. Calling sequence

ANS	Answer (dependent variable)

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

METHOD

$$\text{ANS} = \left\{a_o + \cdot \cdot \cdot \left[a_{n-2} + (a_{n-1} + a_n\, \mathbf{x})\, \mathbf{x}\right] \mathbf{x} \cdot \cdot \cdot\right\}$$

where

ARG = x

CØEFS = $a_o, a_1, \cdot \cdot \cdot, a_{NCØEFS}$

SUBROUTINE IDENTIFICATION

A. Title

PØPPC

B. Segment

ESPØDEPH

C. Called by subroutine

UPDATE

FUNCTION

The function is to compute the matrix which takes a Cartesian covariance matrix into an ECI orbit plane matrix up, down, cross. The dimension of the resulting matrix will either be 6, 7, or 8 depending on the presence or absence of either or both of the drag parameters.

USAGE

A. Calling sequence

Call PØPPC

B. Input

1. CØMMØN

NDPR	Total number of Category 1 variables to solve for
TEMP	Temporary storage
TRAJX(1)	x (e. r.)
TRAJX(2)	y (e. r.)
TRAJX(3)	z (e. r.)
TRAJX(4)	$\dot{x}$ (e. r. /min)
TRAJX(5)	$\dot{y}$ (e. r. /min)
TRAJX(6)	$\dot{z}$ (e. r. /min)

2. Calling sequence

C. Output

1. CØMMØN

TDPDX	Contains matrices of partials for covariance matrix update (either 36, 49 or 62 cells)

D. Error/action messages

—

PØPPC

SUBROUTINES USED

A. Library

SQRTF

B. Program

EQUATIONS

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

$$r \cdot v = x\dot{x} + y\dot{y} + z\dot{z}$$

$$|J| = \sqrt{r^2 v^2 - (r \cdot v)^2}$$

$$|r \times J| = \sqrt{r^2 \left(r^2 v^2 - (r \cdot v)^2\right.}$$

$$\xi_x = \frac{x}{r} \qquad \xi_y = \frac{y}{r} \qquad \xi_z = \frac{z}{R}$$

$$\eta_x = \frac{(r \cdot v)x - r^2 \dot{x}}{|r \times J|} \qquad \eta_y = \frac{(r \cdot v)y - r^2 \dot{y}}{|r \times J|} \qquad \eta_z = \frac{(r \cdot v)z - r^2 \dot{z}}{|r \times J|}$$

$$\zeta_x = \frac{y\dot{z} - z\dot{y}}{|J|} \qquad \zeta_y = \frac{z\dot{x} - x\dot{z}}{|J|} \qquad \zeta_z = \frac{x\dot{y} - y\dot{x}}{|J|}$$

$$[A] = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix}$$

For NDPR = 6

$$TDPDX = U = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}$$

For NDPR = 7 or 8

$$TDPDX = U = \left[\begin{array}{cc|c} A & 0 & 0 \\ 0 & A & 0 \\ \hline 0 & 0 & I \end{array}\right] \left.\vphantom{I}\right\} 1 \text{ or } 2$$

$$\qquad\qquad\qquad\qquad \underbrace{\hphantom{III}}_{1 \text{ or } 2}$$

SUBROUTINE IDENTIFICATION

A. Title

PØSTPR

B. Segment

ESPØDEPH

C. Called by subroutine

MAIN CONTROL

FUNCTION

This subroutine is the driver for the post-processor.

USAGE

A. Calling sequence

Call PØSTPR

B. Input

1. CØMMØN

KØUT	Output tape number
MT	Observation tape number
PSTFLG	ESPØDEPH control flags
TG	Time to integrate to (from 0^h day of epoch)

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

PANT

B. Program

OUTPT	Sets up x_T, y_T, z_T, t_I for punching
RDXYZ	Reads x_T, y_T, z_T, t_I from TTY generated cards
SELECT	Select next time to update to
SETIC	Initialize integration list
TCØMP	Compare $\lvert (x) - x, (y) - y, (z) - z \rvert$ with ϵ
TPRLM	Sets up data for integration
TPRNT	Prints trajectory print
TRAJ	Driver for integration program
TTAPE	Generates ephemeris tape
TWRAP	Wraps up ephemeris tape
UPDATE	Driver for covariance matrix update logic

SUBROUTINE IDENTIFICATION

A. Title

PØTENT

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

DAUX

FUNCTION

Function is to compute the necessary inputs for and to call the GPERT subroutine.

USAGE

A. Calling sequence

Call PØTENT

B. Input

1. CØMMØN

TLIST	Numerical integration working storage
TR	Magnitude of vector from center of earth to vehicle
TALFAG	Right ascension of Greenwich meridian at midnight day of epoch
CWE	Earth's rotation rate (radians/minute)

2. Calling sequence

—

C. Output

1. CØMMØN

SIPH	sin of the geocentric latitude of the vehicle
CØPH	cos of the geocentric latitude of the vehicle
SNALF	sin of the right ascension of the vehicle
CSALF	cos of the right ascension of the vehicle
SILA	sin of the longitude of the vehicle
CØLA	cos of the longitude of the vehicle

2. Calling sequence

—

SUBROUTINES USED

A. Library

CØS

SIN

SQRT

B. Program

GERT

PIMØD

EQUATIONS

$$\cos \phi = \frac{\sqrt{x^2 + y^2}}{R} \qquad \sin \phi = \frac{z}{R}$$

$$\cos a = \frac{x}{\sqrt{x^2 + y^2}} \qquad \sin a = \frac{y}{\sqrt{x^2 + y^2}}$$

$$\lambda = a - (a_{go} + \omega_e t)$$

$$\cos \lambda = \cos a \cos (a_{go} + \omega_e t) + \sin a \sin (a_{go} + \omega_e t)$$

$$\sin \lambda = \sin a \cos (a_{go} + \omega_e t) - \cos a \sin (a_{go} + \omega_e t)$$

SUBROUTINE IDENTIFICATION

A. Title

PPLPC

B. Segment

ESPØDEPH

C. Called by subroutine

UPDATE

FUNCTION

Function is to compute the partial of polar coordinates with respect to Cartesian coordinates and to set up a matrix U necessary to do the update $V = U\Sigma_x U^T$. The dimension of the matrix U will either be 6, 7, or 8 depending on the presence or absence of either or both of the drag parameters.

USAGE

A. Calling sequence

Call PPLPC

B. Input

1. CØMMØN

NDPR		total number of Category 1 variables to solve for
TRAJ	(1)	x (e.r.)
	(2)	y (e.r.)
	(3)	z (e.r.)
	(4)	$\dot{x}$ (e.r./min)
	(5)	$\dot{y}$ (e.r./min)
	(6)	$\dot{z}$ (e.r./min)

2. Calling sequence

—

C. Output

1. CØMMØN

TDPDX Contains matrices of partials for covariance matrix update (either 36, 49, or 64 cells)

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRTF

B. Program

ATNQF Arc tangent

EQUATIONS

$$r^2 = x^2 + y^2 + z^2$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$$

$$r\dot{r} = x\dot{x} + y\dot{y} + z\dot{z}$$

$$\dot{r} = \frac{r\dot{r}}{r}$$

$$q = \frac{1}{r\sqrt{V^2 - \dot{r}^2}}$$

$$A = \tan^{-1}\left(\frac{x\dot{y} - y\dot{x}}{r\dot{z} - z\dot{r}}\right)$$

$$W = \frac{\cos^2 A}{r\dot{z} - z\dot{r}}$$

$$\frac{\partial\alpha}{\partial x} = \frac{-y}{x^2 + y^2} \quad \frac{\partial\alpha}{\partial y} = \frac{x}{x^2 + y^2} \quad \frac{\partial\alpha}{\partial x} = \frac{\partial\alpha}{\partial\dot{x}}\ \frac{\partial\alpha}{\partial\dot{y}}\ \frac{\partial\alpha}{\partial\dot{z}} = 0$$

$$\frac{\partial\delta}{\partial x} = \frac{-xz}{r^2\sqrt{x^2 + y^2}} \quad \frac{\partial\delta}{\partial y} = \frac{-yz}{r^2\sqrt{x^2 + y^2}} \quad \frac{\partial\delta}{\partial z} = \frac{\sqrt{x^2 + y^2}}{r^2} \quad \frac{\partial\delta}{\partial\dot{x}} = \frac{\partial\delta}{\partial\dot{y}}\ \frac{\partial\delta}{\partial\dot{z}} = 0$$

$$\frac{\partial\beta}{\partial x} = q\left(\frac{x\dot{r}}{r} - \dot{x}\right) \quad \frac{\partial\beta}{\partial y} = q\left(\frac{y\dot{r}}{r} - \dot{y}\right) \quad \frac{\partial\beta}{\partial z} = q\left(\frac{z\dot{r}}{r} - \dot{z}\right)$$

$$\frac{\partial\beta}{\partial\dot{x}} = q\left(\frac{\dot{x}r\dot{r}}{V^2} - x\right) \quad \frac{\partial\beta}{\partial\dot{y}} = \left(\frac{\dot{y}r\dot{r}}{V^2} - y\right) \quad \frac{\partial\beta}{\partial\dot{z}} = q\left(\frac{\dot{z}r\dot{r}}{V^2} - z\right)$$

$$\frac{\partial A}{\partial x} = W\left[\dot{y} - \frac{\tan A}{r}\left(\dot{z}x - \dot{x}z + \frac{zx\dot{r}}{r}\right)\right]$$

$$\frac{\partial A}{\partial y} = W\left[-\dot{x} - \frac{\tan A}{r}\left(zy - yz + \frac{zy\dot{r}}{r}\right)\right]$$

$$\frac{\partial A}{\partial z} = \dot{r}\ W \tan A\left(1 - \frac{z^2}{r^2}\right)$$

$$\frac{\partial A}{\partial \dot{x}} = W\left(-y + \frac{xz}{r}\tan A\right)$$

$$\frac{\partial A}{\partial \dot{y}} = W\left(x + \frac{zy}{r}\tan A\right)$$

$$\frac{\partial A}{\partial \dot{z}} = -r\ W \tan A\left(1 - \frac{z^2}{r^2}\right)$$

$$\frac{\partial r}{\partial x} = \frac{x}{r} \quad \frac{\partial r}{\partial y} = \frac{y}{r} \quad \frac{\partial r}{\partial z} = \frac{z}{r} \quad \frac{\partial r}{\partial \dot{x}} = \frac{\partial r}{\partial \dot{y}} = \frac{\partial r}{\partial \dot{z}} = 0$$

$$\frac{\partial v}{\partial x} = \frac{\partial v}{\partial y} = \frac{\partial v}{\partial z} = 0 \quad \frac{\partial v}{\partial \dot{x}} = \frac{\dot{x}}{v} \quad \frac{\partial v}{\partial \dot{y}} = \frac{y}{v} \quad \frac{\partial v}{\partial \dot{z}} = \frac{\dot{z}}{v}$$

$$[A] = \begin{bmatrix} \frac{\partial \alpha}{\partial x} & \frac{\partial \alpha}{\partial y} & \frac{\partial \alpha}{\partial z} & \frac{\partial \alpha}{\partial \dot{x}} & \frac{\partial \alpha}{\partial \dot{y}} & \frac{\partial \alpha}{\partial \dot{z}} \\ \frac{\partial \delta}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial \delta}{\partial \dot{z}} \\ \frac{\partial \beta}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial \beta}{\partial \dot{z}} \\ \frac{\partial A}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial A}{\partial \dot{z}} \\ \frac{\partial r}{\partial x} & \cdot & \cdot & \cdot & \cdot & \frac{\partial r}{\partial \dot{z}} \\ \frac{\partial V}{\partial x} & \frac{\partial V}{\partial y} & \frac{\partial V}{\partial z} & \frac{\partial V}{\partial \dot{x}} & \frac{\partial V}{\partial \dot{y}} & \frac{\partial V}{\partial \dot{z}} \end{bmatrix}$$

PPLPC

For NDPR = 6

$$U = \begin{bmatrix} A \end{bmatrix}$$

For NDPR = 7

$$U = \begin{bmatrix} A & 0 \\ 0 & 1 \end{bmatrix}$$

For NDPR = 8

$$U = \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

SUBROUTINE IDENTIFICATION

A. Title

PPRINT

B. Segment

ESPØDDC

C. Called by subroutine

RADR

FUNCTION

The function is to print residuals information in ESPØDDC.

USAGE

A. Calling sequence

Call PPRINT

B. Input

1. CØMMØN

DCFLG	DC package control flags
DBASE	Number of days from 1950 to day of epoch
IRCNT	Cells for partials print
PDELFG	Cells for partials print
PRESDT	Cells for partials print
PUBS	Sensor number, time R, A, E, Ṙ, α, δ table
TEMP	Temporary storage
CDEG	Degrees/radians
CKMER	km/e.r.
CMTER	Meters/e.r.
KØUT	Output tape number

2. Calling Sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

B. Program

CLTIME	Converts an input time into its Gregorian representation
RMAD	Compares the residual output quantities with a table of maximum values

SUBROUTINE IDENTIFICATION

A. Title

PRAXIS

B. Segment

ESPØDEPH

C. Called by subroutine

UPDATE

FUNCTION

The functions of this subroutine are described below:

a) To compute the eigenvalues and eigenvectors of a real symmetric 3 x 3 matrix, A (stored as a lower triangular matrix). The eigenvectors for the columns of a matrix U and are ordered as column vectors in such a way that the sum of the diagonal elements of the U matrix is maximized.

b) These eigenvectors are then used to compute the three angles ϕ_1, ϕ_2, ϕ_3 which will resolve the matrix A into a diagonal matrix with the eigenvalues of A as the diagonal elements.

USAGE

A. Calling sequence

Call PRAXIS (A, I, B, J)

B. Input

1. CØMMØN

—

2. Calling sequence

L(A)	Address of an array A where the matrix is stored
I	Index to indicate just where in the above array the first element of the matrix is [i.e., A(I) is the first element of the matrix.]
L(B)	Address of an array B where the results of PRAXIS are to be stored
J	Index to indicate just where in the above array the first element of the results are to be stored (See Output for arrangement of results in array B.)

C. Output

1. CØMMØN

—

2. Calling sequence

$$\left.\begin{array}{ll} B(J) & - \lambda_1 \\ B(J+1) & - \lambda_2 \\ B(J+2) & - \lambda_3 \end{array}\right\} \text{eigenvalues of A}$$

$$\left.\begin{array}{ll} B(J+3) & - U_{11} \\ B(J+4) & - U_{12} \\ B(J+5) & - U_{13} \end{array}\right\} \text{first eigenvector}$$

$$\left.\begin{array}{ll} B(J+6) & - U_{21} \\ B(J+7) & - U_{22} \\ B(J+8) & - U_{23} \end{array}\right\} \text{second eigenvector}$$

$$\left.\begin{array}{ll} B(J+9) & - U_{31} \\ B(J+10) & - U_{32} \\ B(J+11) & - U_{33} \end{array}\right\} \text{third eigenvector}$$

$$\left.\begin{array}{ll} B(J+12) & - \phi_1 \\ B(J+13) & - \phi_2 \\ B(J+14) & - \phi_3 \end{array}\right\} \text{rotational angles (rad)}$$

$$\left.\begin{array}{ll} B(J+15) & - \sqrt{\lambda_1} \\ B(J+16) & - \sqrt{\lambda_2} \\ B(J+17) & - \sqrt{\lambda_3} \end{array}\right\} \text{square roots of the three eigenvalues}$$

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRTF

CØSF

SINF

B. Program

ATNQF	Arctangent routine
XCRØSS	Cross product routine

EQUATIONS

Compute the eigenvalues of A

$$m = \frac{1}{3}\,\mathrm{tr}\,(A) \text{ where } \mathrm{tr}(A) = \sum_{i=1}^{3} a_{ii}$$

$$q = \frac{1}{2}\det\,(A - m\,I)$$

6p = sum of the squares of the elements of (A - mI). From "Cardano's" trigonometric solution of det $\left[(A - mI) - \mu I\right]$ as a cubic in μ, the eigenvalues of A are

$$\lambda_1 = m + 2\sqrt{p}\cos\phi$$

$$\lambda_2 = m - \sqrt{p}\,(\cos\phi + \sqrt{3}\sin\phi)$$

$$\lambda_3 = m + \sqrt{p}\,(\cos\phi - \sqrt{3}\sin\phi)$$

where

$$\phi = \frac{1}{3}\tan^{-1}\frac{\sqrt{p^3 - q^2}}{q} \qquad 0 \leq \phi \leq \frac{\pi}{3}$$

Compute the eigenvectors. Let λ represent one of the three eigenvalues λ_1, λ_2, λ_3.

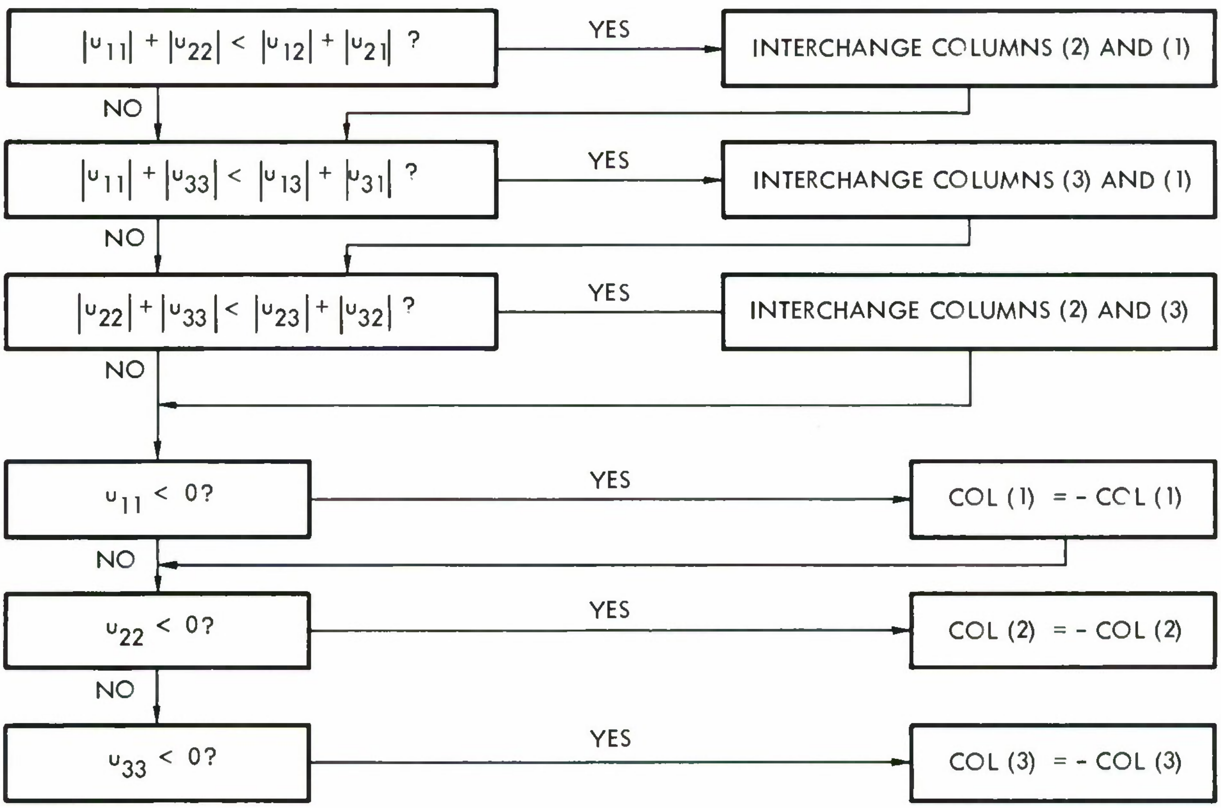
$$\vec{C}_1 = \begin{pmatrix} a_{11} - \lambda \\ a_{21} \\ a_{31} \end{pmatrix} \times \begin{pmatrix} a_{21} \\ a_{22} - \lambda \\ a_{32} \end{pmatrix}$$

$$\vec{C}_2 = \begin{pmatrix} a_{21} \\ a_{22} - \lambda \\ a_{32} \end{pmatrix} \times \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} - \lambda \end{pmatrix}$$

$$\vec{C}_3 = \begin{pmatrix} a_{31} \\ a_{32} \\ a_{33} - \lambda \end{pmatrix} \times \begin{pmatrix} a_{11} - \lambda \\ a_{21} \\ a_{31} \end{pmatrix}$$

If $\vec{C}_1 \cdot \vec{C}_2 < 0$; set $\vec{C}_2 = -\vec{C}_2$. If $\vec{C}_1 \cdot \vec{C}_3 < 0$; set $\vec{C}_3 = -\vec{C}_3$. $\vec{u} = 1/3\ (\vec{C}_1 + \vec{C}_2 + \vec{C}_3)$. $\vec{u} = \vec{u}/|\vec{u}|$ is the eigenvector corresponding to λ.

Letting the three eigenvectors from the columns of the matrix U, the following diagram shows the logic used to maximize the sum of the diagonal elements of U.



PRAXIS

Finally, compute ϕ_1, ϕ_2, ϕ_3

$$\phi_1 = \tan^{-1}\left[-\frac{u_{23}}{u_{22}}\right]$$

$$\phi_2 = \sin^{-1}\left[-u_{21}\right]$$

$$\phi_3 = \tan^{-1}\left[-\frac{u_{31}}{u_{11}}\right]$$

SUBROUTINE IDENTIFICATION

A. Title

PRCØNS

B. Segment

ESPØD

C. Called by subroutine

IPRNT

FUNCTION

The functions are to print the program constants, the sensor types, and to print the sensor sigmas.

USAGE

A. Calling sequence

Call PRCØNS

B. Input

1. CØMMØN

CSIG	Sixty sets of sensor sigmas
CSTYPE	Sensor sigmas for σ, $\bar{N}_S$, and N
TEMP	Temporary storage

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

PANT

B. Program

UNPAKSN   Unpacks the sensor sigmas

SUBROUTINE IDENTIFICATION

A. Title

PRECES

B. Segment

ESPØD

C. Called by subroutine

SWTSN

FUNCTION

This subroutine sets up information for the ADJUST subroutine. Testing is done to determine if the data is field-reduced or precision-reduced, and to determine the year in which the data is referenced. (See card format.)

USAGE

A. Calling sequence

Call PRECES (A, B, C)

B. Input

1. CØMMØN

—

2. Calling sequence

A Declination (rad)
B Right ascension (rad)
C Equipment type and equinox (packed)

C. Output

1. CØMMØN

—

2. Calling sequence

A Declination, precessed (rad)
B Right ascension, precessed (rad)

D. Error/ action messages

—

SUBROUTINES USED

A. Library

—

B. Program

ADJUST Updates right ascension, declination measurements to equinox of integration

CARD FORMAT

If column 80 = Δ do not precess

If equipment type = 16. (field reduced)

If equipment type = 17. (precision reduced)

If column 4, 5 = 16 and $\delta \geq -23^\circ$ equinox year = 1855

If column 4, 5 = 16 and $\delta < -23^\circ$ equinox year = 1875

Otherwise column 70 = 0 equinox year of 1963

Otherwise column 70 = 1 equinox year of 1900

Otherwise column 70 = 2 equinox year of 1925

Otherwise column 70 = 3 equinox year of 1950

Otherwise column 70 = 4 equinox year of 1975

Otherwise column 70 = 5 equinox year of 2000

Otherwise column 70 = 6 equinox year of 1850

Otherwise column 70 = 7 equinox year of 1855

Otherwise column 70 = 8 equinox year of 1875

Otherwise column 70 = 9 equinox year of 1960

Otherwise column 70 = 10 do not precess

SUBROUTINE IDENTIFICATION

A. Title

PRELIM

B. Segment

ESPØDDC

C. Called by subroutines

RADR

FUNCTION

The function is to calculate preliminary quantities for the formulation of residuals and partial derivatives of observation with respect to solution parameters.

USAGE

A. Calling sequence

Call PRELIM

B. Input

1. CØMMØN

a.	PSTAT(4)	Cos ϕ^*
	PSTAT(5)	Sin ϕ^*
	PSTAT(6)	$\alpha g_0 + \lambda$ (rad)
	PSTAT(7)	w_1^s (e. r.)
	PSTAT(8)	w_3^s (e. r.)
b.	PUBS(1)	T (min)
	PUBS(6)	$\dot{R}$ (e. r. /min)
c.	TRAJ(1)	x
	TRAJ(2)	y
	TRAJ(3)	z
	TRAJ(4)	$\dot{x}$
	TRAJ(5)	$\dot{y}$
	TRAJ(6)	$\dot{z}$
	TRAJ(10)	> TRAJX(57) = partials of TRAJ(1-6) with respect to Pi, i = 1, NDPR
d.	NDPR	Number of all differential plus initial parameters to solve for (Category 1)
e.	TEMP	Temporary storage
f.	CWE	Earth's rotational rate

2. Calling sequence

—

C. Output

1. CØMMØN

a.	PCMA	R = computed slant range
b.	PCSA	$\cos A_c$
c.	PCSALF	$\cos(\alpha_g)$
d.	PCSE	$\cos E_c$
e.	PRSUB1	$\dot{R}_1 = V_R$
f.	PSNA	$\sin A_c$
g.	PSNALF	$\sin(\alpha_g)$
h.	PSNE	$\sin E_c$
i.	PUDTI	Vector $(\dot{u}_1, \dot{u}_2, \dot{u}_3)$
j.	PUI	Vector (u_1, u_2, u_3)
k.	PV	$\sqrt{v_1^2 + v_2^2}$
l.	PVI	Vector (v_1, v_2, v_3)
m.	PWDTI	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
n.	PWDTPP	Partial derivatives
o.	PWI	Vector (w_1, w_2, w_3)
p.	PWPP	Partial derivatives

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF
SINF
SQRTF

B. Program

—

EQUATIONS

The computed orbit positions (x, y, z) and station positions (ϕ^*, λ, h) are processed to produce geocentric and topcentric coordinates of the vehicle in an Earth-fixed coordinate system. Right ascensions of the station for times of observations t_i are

$$\alpha_i = (\alpha_{go} + \lambda) + \omega_e (t_i - t_o)$$

Geocentric position and velocity of the vehicle in Earth-fixed coordinates are

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_i = \begin{bmatrix} \cos \alpha_i & \sin \alpha_i & 0 \\ -\sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} \dot{w}_1 \\ \dot{w}_2 \\ \dot{w}_3 \end{bmatrix}_i \quad \begin{bmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} + \omega_e y \\ \dot{y} - \omega_e x \\ \dot{z} \end{bmatrix}$$

The station position in meridian coordinates is provided by the preprocessor module where it is computed from geodetic latitude, ϕ^*, and altitude, h, as follows.

$$A_s = \left(\cos^2 \phi^* + b_e^2 \sin^2 \phi^* \right)^{-1/2}$$

$$B_s = \left(\sin^2 \phi^* + \frac{1}{b_e^2} \cos^2 \phi^* \right)^{-1/2}$$

$$w_1^s = (A_s + h) \cos \phi^*$$

$$w_3^s = (b_e B_s + h) \sin \phi^*$$

where b_e is the polar axis of the reference spheroid.

Topocentric coordinates, direction cosines and related quantities for the vehicle in meridian plane coordinate system are then

$q_1 = w_1 - w_1^s$ (Topocentric position in equatorial coordinate system)

$q_2 = w_2$

$$q_3 = w_3 - w_3^s$$

$$R = \sqrt{q_1^2 + q_2^2 + q_3^2}$$

$$\bar{u} = \begin{cases} u_1 = q_1/r \\ u_2 = q_2/r \\ u_3 = q_3/r \end{cases}$$

(Topocentric direction cosines in equatorial system)

$$\dot{\bar{u}} = \begin{cases} \dot{u}_1 = (\dot{w}_1 - K\,u_1)/r \\ \dot{u}_2 = (\dot{w}_2 - K\,u_2)/r \\ \dot{u}_3 = (\dot{w}_3 - K\,u_3)/r \end{cases}$$

$$K = u_1\,\dot{w}_1 + u_1\,\dot{w}_2 + u_3\,\dot{w}_3$$

$$\bar{v} = \begin{cases} v_1 = u_2 \\ v_2 = -u_1 \sin\phi^* + u_3 \cos\phi^* \\ v_3 = u_1 \cos\phi^* + u_3 \sin\phi^* \end{cases}$$

(Topocentric direction cosines in horizon system)

$$V = \sqrt{v_1^2 + v_2^2}$$

$$R_1 = VR$$

$$\sin E = v_3$$

$$\cos E = V$$

$$\cos A = v_2/V$$

$$\sin A = v_1/V$$

$$\begin{bmatrix} \dfrac{\partial w_1}{\partial p_i} \\ \\ \dfrac{\partial w_2}{\partial p_i} \\ \\ \dfrac{\partial w_3}{\partial p_i} \end{bmatrix} = \begin{bmatrix} \cos a & \sin a & 0 \\ \\ -\sin a & \cos a & 0 \\ \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{\partial x}{\partial p_i} \\ \\ \dfrac{\partial y}{\partial p_i} \\ \\ \dfrac{\partial z}{\partial p_i} \end{bmatrix}$$

If range rate observations are used (PUBS $\neq 0$), then variational equations in velocity are rotated as follows.

$$\begin{bmatrix} \dfrac{\partial \dot{w}_1}{\partial p_i} \\ \\ \dfrac{\partial \dot{w}_2}{\partial p_i} \\ \\ \dfrac{\partial \dot{w}_3}{\partial p_i} \end{bmatrix} = \begin{bmatrix} \cos a & \sin a & 0 \\ \\ -\sin a & \cos a & 0 \\ \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{\partial \dot{x}}{\partial p_i} + \omega_e \dfrac{\partial y}{\partial p_i} \\ \\ \dfrac{\partial \dot{y}}{\partial p_i} - \omega_e \dfrac{\partial x}{\partial p_i} \\ \\ \dfrac{\partial \dot{z}}{\partial p_i} \end{bmatrix}$$

where the parameters p_i are the ADBARV conditions at epoch (α_o, δ_o, β_o, A_o, r_o, v_o), drag parameter ($C_D A/2m$) and coefficient of diurnal drag variation, ϵ.

SUBROUTINE IDENTIFICATION

A. Title

PRSSTB

B. Segment

ESPØDDC

C. Called by subroutine

INTEG

FUNCTION

The function is to print the table containing estimates of the means and standard deviations of residuals by sensor and type.

USAGE

A. Calling sequence

Call PRSSTB

B. Input

1. CØMMØN

VSTR (NSSTB) Each sensor with data involved in the differential correction is represented in this array in the following format:

Sensor No.

$\sum \Delta R_i \qquad i = 1, — N_R^A$

$\sum \Delta R_i^2$

$N_R^A * 10000 + N_R^R$

$\sum \Delta A_i \qquad i = 1, — N_A^A$

$\sum (\Delta A_i)^2$

$N_A^A * 10000 + N_A^R$

$\sum \Delta E_i \qquad i = 1, — N_E^A$

$\sum (\Delta E_i)^2$

$$N_E^{\;A} * 10000 + N_E^{\;R}$$

$$\sum \Delta \dot{R}_i \qquad\qquad i = 1, — N_{RDT}^{R}$$

$$\sum (\Delta \dot{R}_i)^2$$

$$N_{RDT}^{A} * 10000 + N_{RDT}^{R}$$

TEMP Temporary storage
KØUT Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library
SQRTF

B. Program

—

EQUATIONS

$$M_R = \frac{\sum \Delta R_i)}{N_R^{\;A}} , \qquad \sigma_{\Delta R} = \sqrt{\frac{\sum (\Delta R_i)^2}{N_R^{\;A}} - M_R^2}$$

$$M_A = \frac{\sum (\Delta A_i)}{N_A^{\;A}} , \qquad \sigma_{\Delta A} = \sqrt{\frac{\sum (\Delta A_i)^2}{N_A^{\;A}} - M_A^2}$$

$$M_E = \frac{\sum(\Delta E_i)}{N_E^A} \quad , \qquad \sigma_{\Delta E} = \sqrt{\frac{\sum(\Delta E_i)^2}{N_E^A} - M_E^2}$$

$$M_{RDT} = \frac{\sum(\Delta RDT_i)}{N_{RDT}^A} \quad , \qquad \sigma_{\Delta RDT} = \sqrt{\frac{\sum(\Delta RDT_i)^2}{N_{RDT}^A} - M_R^2}$$

SUBROUTINE IDENTIFICATION

A. Title

PTØC

B. Segment

ESPØD

ESPØDDC

C. Called by subroutine

DPRLM (ESPØD)

APPLY (ESPØDDC)

FUNCTION

The function is to convert polar coordinates to Cartesian coordinates.

USAGE

A. Calling sequence

Call PTØC (C, D)

B. Input

1. CØMMØN

—

2. Calling sequence

C The address of a six-dimensional array containing α, δ, β, A, R, V (angles in radians)

C. Output

1. CØMMØN

—

2. Calling sequence

D The address of a six-dimensional array which will contain x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$. The units of this set will be the same as the input R, v units.

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF

SINF

EQUATIONS

$$x = R \cos \delta \cos \alpha$$

$$y = R \cos \delta \sin \alpha$$

$$z = R \sin \delta$$

$$\dot{x} = v\left[\cos \alpha(-\cos A \sin \beta \sin \delta + \cos \beta \cos \delta) - \sin A \sin \beta \sin \alpha\right]$$

$$\dot{y} = v\left[\sin \alpha(-\cos A \sin \beta \sin \delta + \cos \beta \cos \delta) + \sin A \sin \beta \cos \alpha\right]$$

$$\dot{z} = v(\cos A \cos \delta \sin \beta + \cos \beta \sin \delta)$$

SUBROUTINE IDENTIFICATION

A. Title
PUPB

B. Segment
ESPØDDC

C. Called by subroutine
RADR

FUNCTION

The function of this subroutine is to evaluate the partials of observations with respect to biases of time, sensor latitude, sensor longitude, and sensor altitude. The observation type and the bias type are given in the calling sequence.

USAGE

A. Calling sequence
PUPB (I, J)

B. Input

1. CØMMØN

CØUNT	Number of lines
PCMR	R = computed slant range
PCSA	Cos A_c
PCSALF	Cos (α_g)
PCSE	Cos E_c
PRSUB1	R_1 = VR
PSNA	Sin A_c
PSNALF	Sin (α_g)
PSNE	Sin E_c
PSTAT	Working storage for sensor information
PUDTI	Vector $(\dot{u}_1, \dot{u}_2, \dot{u}_3)$
PUI	Vector (u_1, u_2, u_3)

PV	$\sqrt{V_1^2 + V_2^2}$
PVI	Vector (V_1, V_2, V_3)
PWDTI	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
PWI	Vector (w_1, w_2, w_3)
TRAJX	$x, y, z, \dot{x}, \dot{y}, \dot{z} \cdots$
CWE	Earth's rotational rate (rad/min)
KØUT	Output tape number

2. Calling sequence

J = 1 for R

= 2 for A

= 3 for E

= 4 for $\dot{R}$

= 5 for H

= 6 for D

I = 7 for t_b

= 8 for ϕ_b^*

= 9 for ℓ_b

= 10 for h_b

C. Output

1. CØMMØN

—

2. Calling sequence

$$\text{A register} = \frac{\partial(\text{variable J})}{\partial(\text{variable I})}$$

D. Error/action messages

PARTIAL ( ) WITH RESPECT ( ) ASKED FOR

Given off-line when I and J exceed current program limits (I = 10, J = 6)

SUBROUTINES USED

A. Library

GLØP

B. Program

LINES Line counter

EQUATIONS

Range (type 1 observation)

$$\frac{\partial R}{\partial \phi^*} = u_1 w_3^s - u_3 w_1^s \text{ (type 8 bias)}$$

$$\frac{\partial R}{\partial \lambda} = u_1 w_2 - u_2 w_1 \text{ (type 9 bias)}$$

$$\frac{\partial R}{\partial h} = -u_1 \cos \phi^* - u_3 \sin \phi^* \text{ (type 10 bias)}$$

$$\frac{\partial R}{\partial t} = u_1 \dot{w}_1 + u_2 \dot{w}_2 + u_3 \dot{w}_3 \text{ (type 7 bias)}$$

Azimuth (type 2 observation)

$$\frac{\partial A}{\partial \phi^*} = \frac{\sin A}{R_1} (w_1 \cos \phi^* + w_3 \sin \phi^*) \text{ (type 8 bias)}$$

$$\frac{\partial A}{\partial \lambda} = \frac{w_1 \cos A + w_2 \sin \phi^* \sin A}{R_1} \text{ (type 9 bias)}$$

$$\frac{\partial A}{\partial h} = 0 \text{ (type 10 bias)} \quad R_1 = VR$$

$$\frac{\partial A}{\partial t} = \frac{1}{V^2} (v_2 \dot{v}_1 - v_1 \dot{v}_2) \text{ (type 7 bias)}$$

Elevation (type 3 observation)

$$\frac{\partial E}{\partial \phi^*} = \frac{1}{R_1} \left(w_3 \cos \phi^* - w_1 \sin \phi^* - \frac{\partial R}{\partial \phi^*} \sin E \right) \text{ (type 8 bias)}$$

$$\frac{\partial E}{\partial \lambda} = \frac{1}{R_1} \left(w_2 \cos \phi^* - \frac{\partial R}{\partial \lambda} \sin E \right) \text{ (type 9 bias)}$$

$$\frac{\partial E}{\partial h} = -\frac{1}{R_1}\left(1 + \frac{\partial R}{2h} \sin E\right) \text{(type 10 bias)}$$

$$\frac{\partial E}{\partial t} = \frac{\dot{u}_1 \cos \phi^* + \dot{u}_3 \sin \phi^*}{\cos E} \quad \text{(type 7 bias)}$$

Range Rate (type 4 observation)

$$\frac{\partial \dot{R}}{\partial \phi^*} = w_3{}^s \dot{u}_1 - w_1{}^s \dot{u}_3 \text{ (type 8 bias)}$$

$$\frac{\partial \dot{R}}{\partial \lambda} = (w_2\dot{u}_1 - w_1\dot{u}_2) + (\dot{w}_2 u_1 - \dot{w}_1 u_2) \text{ (type 9 bias)}$$

$$\frac{\partial \dot{R}}{\partial h} = -\dot{u}_1 \cos \phi^* - \dot{u}_3 \sin \phi^* \text{ (type 10 bias)}$$

$$\frac{\partial \dot{R}}{\partial t} = \ddot{R} = (\ddot{\overline{w}} \cdot \overline{u}) + \frac{\|\dot{\overline{w}}\|}{R} - \frac{(\dot{\overline{w}} \cdot \dot{\overline{u}})^2}{R} \quad \text{(type 10 bias)}$$

where

$$\ddot{\overline{w}} = \begin{cases} \ddot{w}_1 = -\omega_e{}^2 w_1 + 2\omega_e(-\dot{x} \sin a + \dot{y} \cos a) + (\ddot{x} \cos a + \ddot{y} \sin a) \\ \ddot{w}_2 = -\omega_e{}^2 w_2 + 2\omega_e(-\dot{x} \cos a - \dot{y} \sin a) + (-\ddot{x} \sin a + \ddot{y} \cos a) \\ \ddot{w}_3 = \ddot{z} \end{cases}$$

Hour Angle (type 5 observation)

$$\frac{\partial H}{\partial \phi^*} = \cos^2 H\left(\frac{q_2}{q_1{}^2}\right) w_3{}^s \quad \text{(type 8 bias)}$$

$$\frac{\partial H}{\partial \lambda} = \cos^2 H\left(\frac{q_1 w_1 + q_2 w_2}{q_1{}^2}\right) \quad \text{(type 9 bias)}$$

$$\frac{\partial H}{\partial h} = \cos^2 H \cos \phi^* \frac{q_2}{q_1} \quad \text{(type 10 bias)}$$

$$\frac{\partial H}{\partial t} = -\left(\frac{\dot{u}_1 \sin H + \dot{u}_2 \cos H}{\cos D}\right) \text{(type 7 bias)}$$

Declination (type 6 observation)

$$\frac{\partial D}{\partial \phi^*} = \frac{-w_1^s - \frac{\partial R}{\partial \phi^*} \sin D}{R \cos D} \quad \text{(type 8 bias)}$$

$$\frac{\partial D}{\partial \lambda} = \frac{\partial R}{\partial \lambda} \frac{\sin D}{R \cos D} \quad \text{(type 9 bias)}$$

$$\frac{\partial D}{\partial h} = \frac{-\sin \phi^* - \frac{\partial R}{\partial h} \sin D}{R \cos D} \quad \text{(type 10 bias)}$$

$$\frac{\partial D}{\partial D_{bias}} = 1 \quad \text{(type 6 bias)}$$

$$\frac{\partial D}{\partial t} = \frac{\dot{u}_3}{\cos D} \quad \text{(type 7 bias)}$$

SUBROUTINE IDENTIFICATION

A. Title

RADR

B. Segment

ESPØDDC

C. Called by subroutine

INTEG

FUNCTION

Function is to control region for the formulation of the system of equations to be solved (Ax = B). A is the matrix of partial derivatives of observations with respect to solution variables and B is the vector of observation residuals. RADR also drives those routines which, given A, B, form $A^T A$, $A^T B$, and $B^T B$. It also drives the residuals print routines.

USAGE

A. Calling sequence

Call RADR

B. Input

1. CØMMØN

CKRMS	Sigma multiplier for deletion criterion
CØUNT	Number of lines
IPFRST	0 to indicate first time in RADR
IRCNT	Cells for partials print
IVSTR	Fixed point variable storage
NARØW	Starting location where one row of the augmented matrix (A, B) is stored
NDPR	Number of all differential plus initial parameters to solve for (Category 1)
NPBIS	Identifies table for current estimates of Category 2 variables
NPR	Number of all parameters to solve for
NPRCD	Identifies table for definition of Category 2 variables to be solved for

NSSTB	Identifies the starting location where station information concerning computed sigmas and means of residuals are stored
PCMR	Computed slant range
PDELFG	Cells for partials print
PØBCNT	Total number of accepted observations
PRESD	Residuals
PSIG	Sigma list
PSTAT	Working storage for sensor information
PUBS	Sensor number, time, R, A E, $\dot{R}$, α, δ table
PUI	Vector (u_1, u_2, u_3)
PVI	Vector (v_1, v_2, v_3)
PWDTI	Vector $(\dot{w}_1, \dot{w}_2, \dot{w}_3)$
TSUS	Current total SØS
VSTR	Floating point variable storage
CPI	π
C2PI	2π

2. Calling sequence

—

C. Output

1. CØMMØN

The array VSTR (NATA) contains the total A^TA, A^TB, B^TB.

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

ASIN	Arc sine routine
ATNQF	Arc tangent routine
DRDP	Partials of observations w.r.t. Category 1 variables

LEGS1	Forms A^TA and A^TB given A and B
LINES	Line counter
PARØUT	Computes additional residual information
PASTØR	Sets up an asterisk for printing
PHEAD	Prints residuals header
PIMØD	Principal value of angle between 0 and 2π
PPRINT	Prints residuals
PRELIM	Preliminary calculations
PUPB	Partials of observations w.r.t. Category 2 variables
SSTB	Accumulates sum, sum of squares, and number of residuals by sensor and data type

EQUATIONS

Computation of Observables from Fitted Orbit

The fitted orbit is used to produce computed "observables" for comparison with observations.

$$R = \sqrt{q_1^2 + q_2^2 + q_3^2} \qquad \text{(range)}$$

$$A = \tan^{-1} v_1/v_2 \qquad \text{(azimuth)}$$

$$E = \sin^{-1} v_3 = \cos^{-1} V \qquad \text{(elevation)}$$

$$H = \tan^{-1} (-u_2/u_1) \qquad \text{(topocentric local hour angle)}$$

$$D = \sin^{-1} u_3 \qquad \text{(topocentric declination)}$$

$$R = \overline{u} \cdot \overline{W} \qquad \text{(range rate)}$$

Computation of partial derivatives of observations with respect to observational biases is shown below.

$$\frac{\partial \phi_j}{\partial \phi_{b_i}} = \begin{cases} 1. \text{ if } i = j & i = 1, 2, 3, 4, 5, 6 \\ 0. \text{ if } i \neq j & j = 1, 2, 3, 4, 5, 6 \end{cases}$$

SUBROUTINE IDENTIFICATION

A. Title

READPR

B. Segment

ESPØD

C. Called by subroutine

JDCSRCH (ESPØD)
MAIN (ESPØD)

FUNCTION

This subroutine reads input cards in the specified formats. A diagnostic check is made to determine if certain conditions are satisfied. If an input error condition occurs, the next JDC card will be read and then the cards for the case will be processed.

RDØNE is a self-contained subroutine within READPR, referenced from outside. This routine reads one card and returns with the card image location in index register 3.

SIGPACK is a self-contained subroutine within READPR referenced from outside. The routine takes four floating point numbers, scales them by 10^4 and packs them as two words in the specified location. The packing is that specified for CSIG table.

USAGE

A. Calling sequence

Call READPR

B. Input

CDRUNB	Running buffer count for card input
CARBUF	Card input buffer
CLDSTR	Cold start flag
FRSTFL	First time flag
CØNVR	Conditional start flag
ICTYP	Used to test need for ICØND and ITIME cards

C. Output

Cells are filled depending on the name of the identification field as well as the card number in columns 1 and 2.

Flags are set according to certain type of cards which have been read. These flags are then analyzed before returning control to the calling program.

D. Error/action messages

1. On and Off line

DATA NAME ( ) NOT FOUND.

Card name is illegal, proceed with reading cards.

CONDITIONAL REQUIREMENTS NOT MET—PROCEEDING TO NEXT CASE.

CLDSTR (cold start flag) = 2 and CØNVR ≠ 0, find next JDC card and then start reading more data.

2. Off Line

( ) CARD REQUIRED FOR THIS RUN.

(ICTYP) ICØND and ITIME are in the case, no ICTYP card.

(ICØND) ICTYP and ITIME cards are in the case, no ICØND card.

(ITIME)-ICTYP and ICØND card are in this case, no ITIME card.

COLUMN 68 IN CARD 7 IS ILLEGAL. CARD IMAGE BELOW.

Column 68 in element card 7 is not a 0, 1 or 2; proceed with reading cards.

SUBROUTINES USED

A. Library

XFIX	Convert to fixed Fortran integer
MXØRD	Read data tape

B. Program

FLEX	Write comments on typewriter
GLØP	Write output tape 11
JDCSRCH	Search for next JDC card
XSRCH	Read card images
PANT	Spacing control

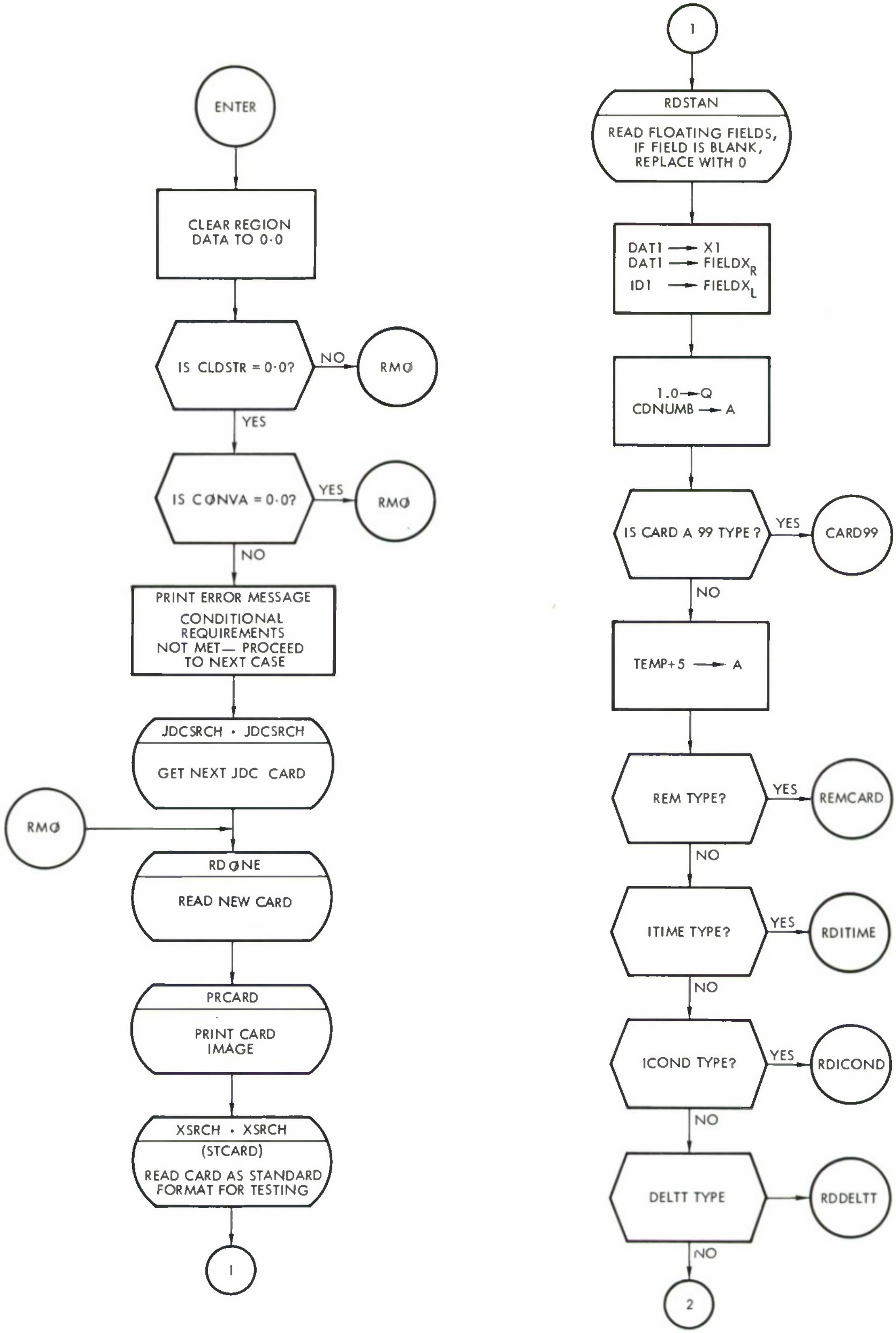


Figure 4-7 a. READPR Flow Diagram

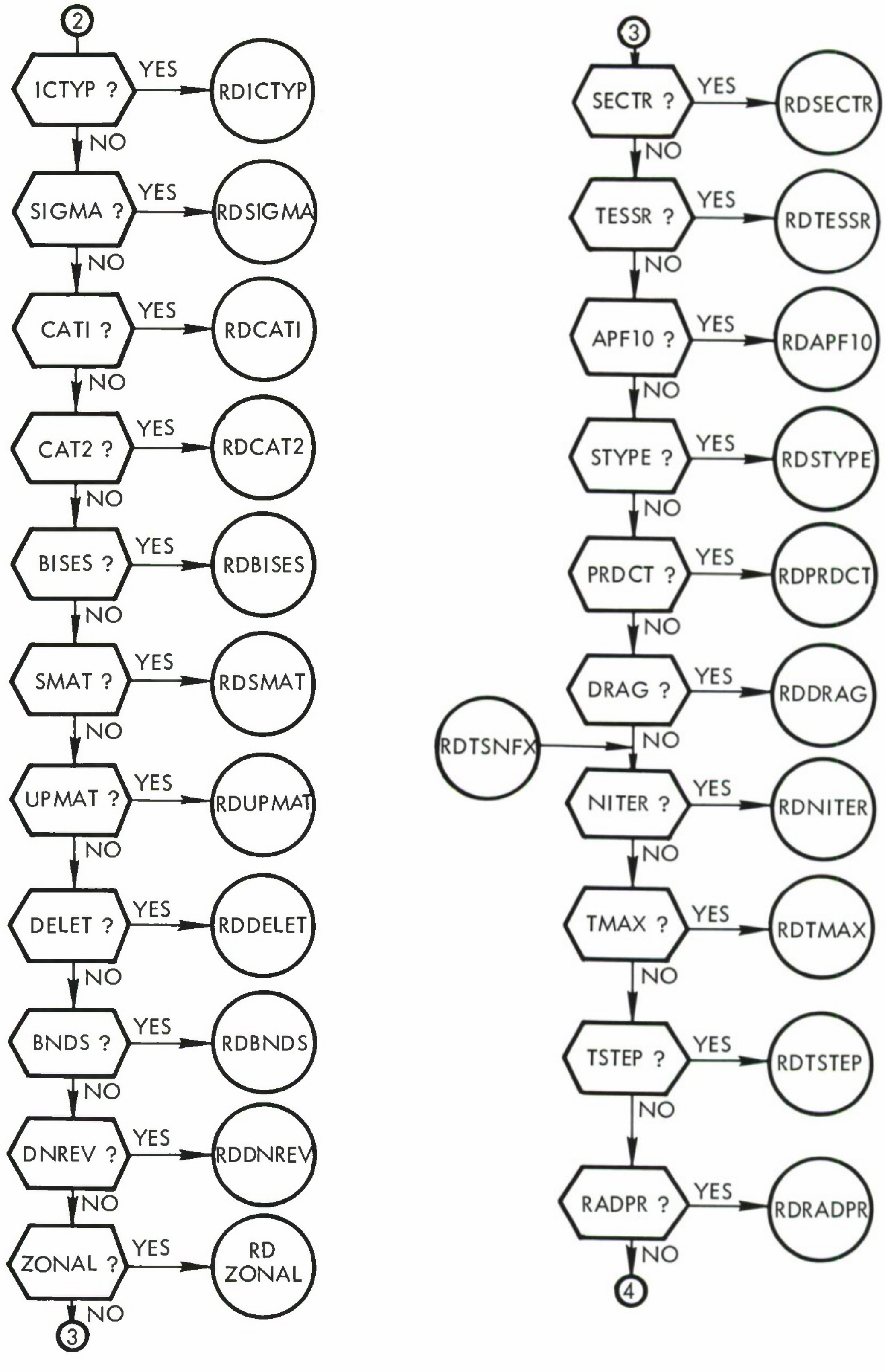


Figure 4-7 b. READPR Flow Diagram (Continued)

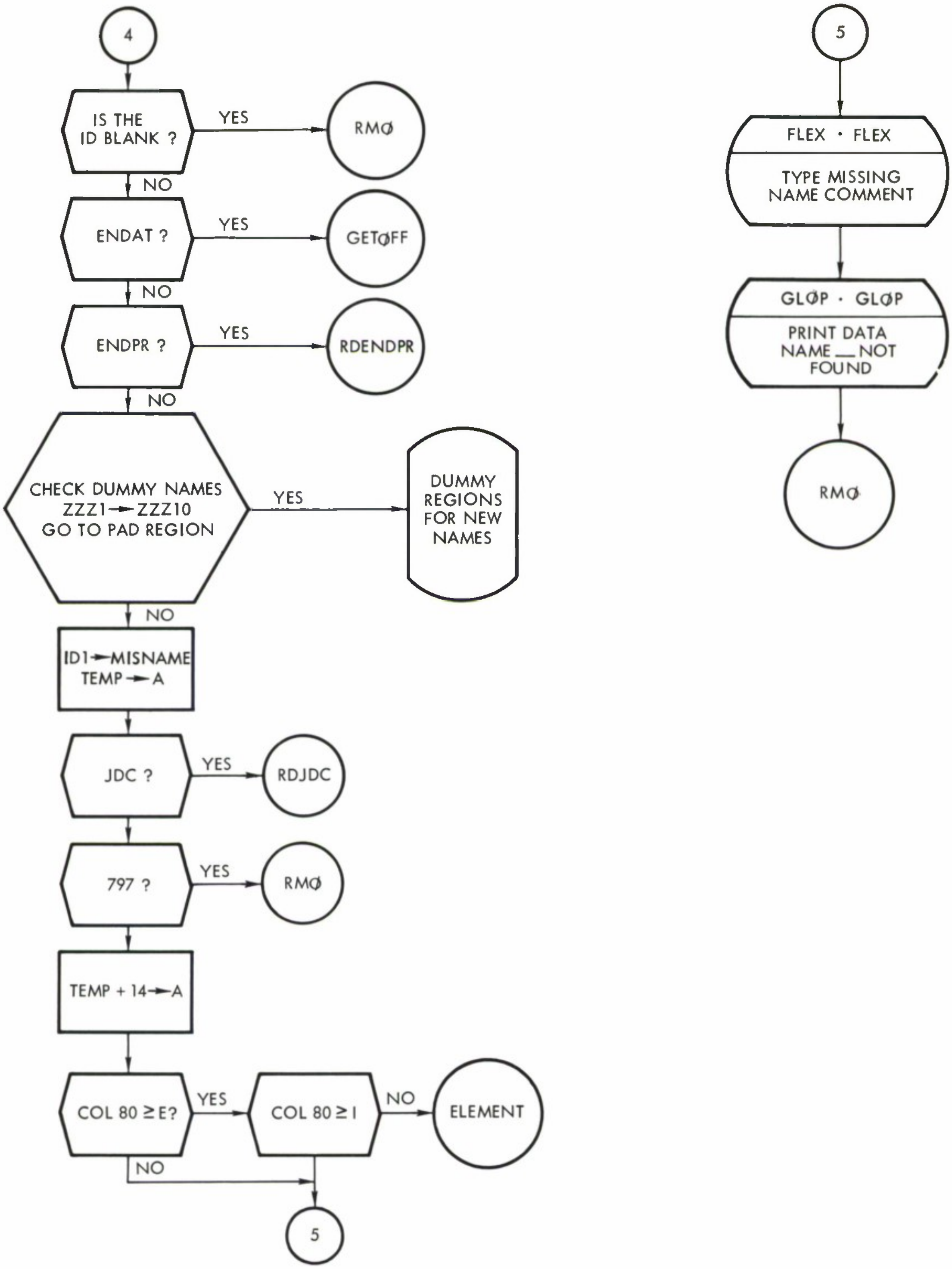


Figure 4-7 c. READPR Flow Diagram (Continued)

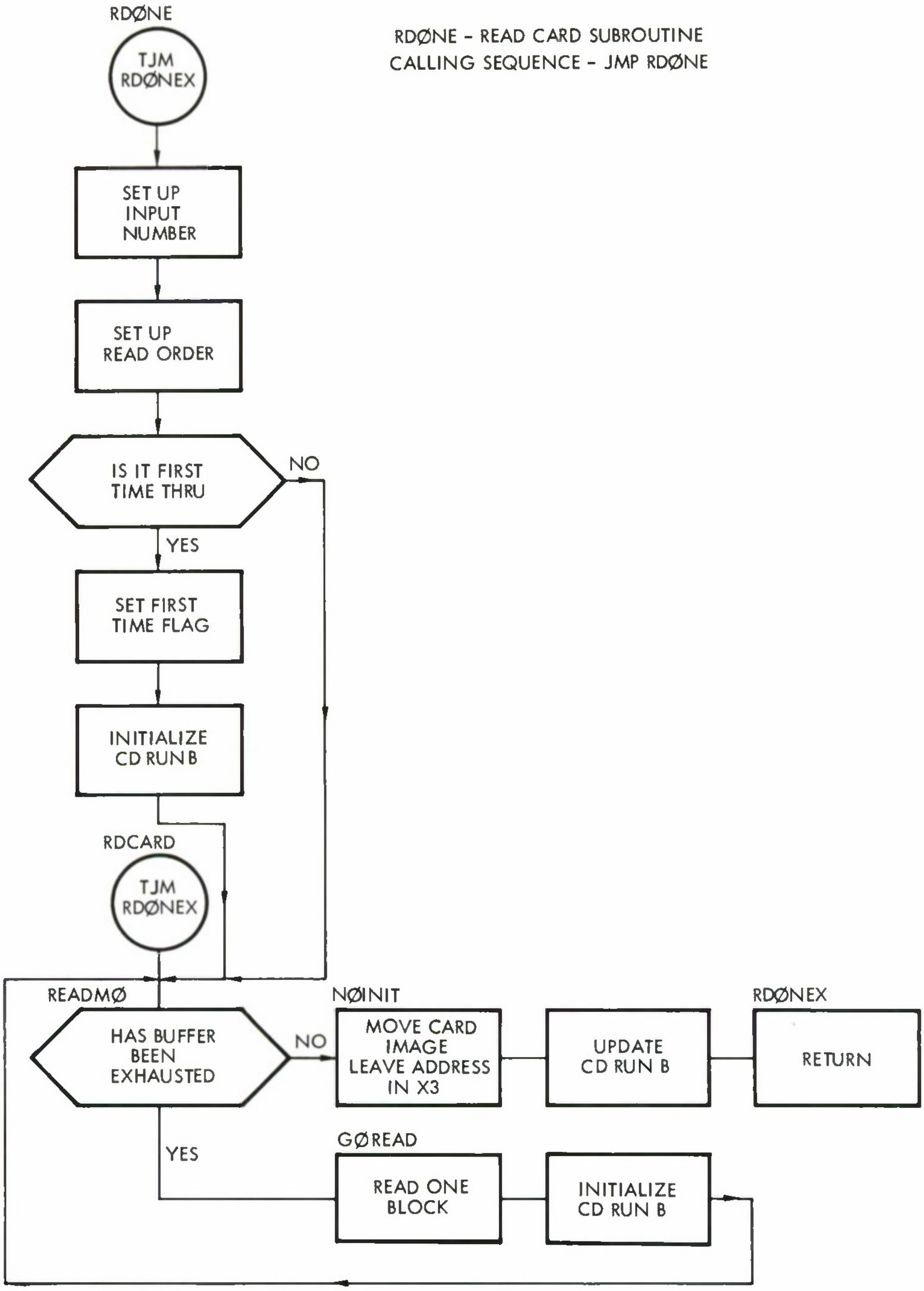


Figure 4-8. RDØNE Flow Diagram

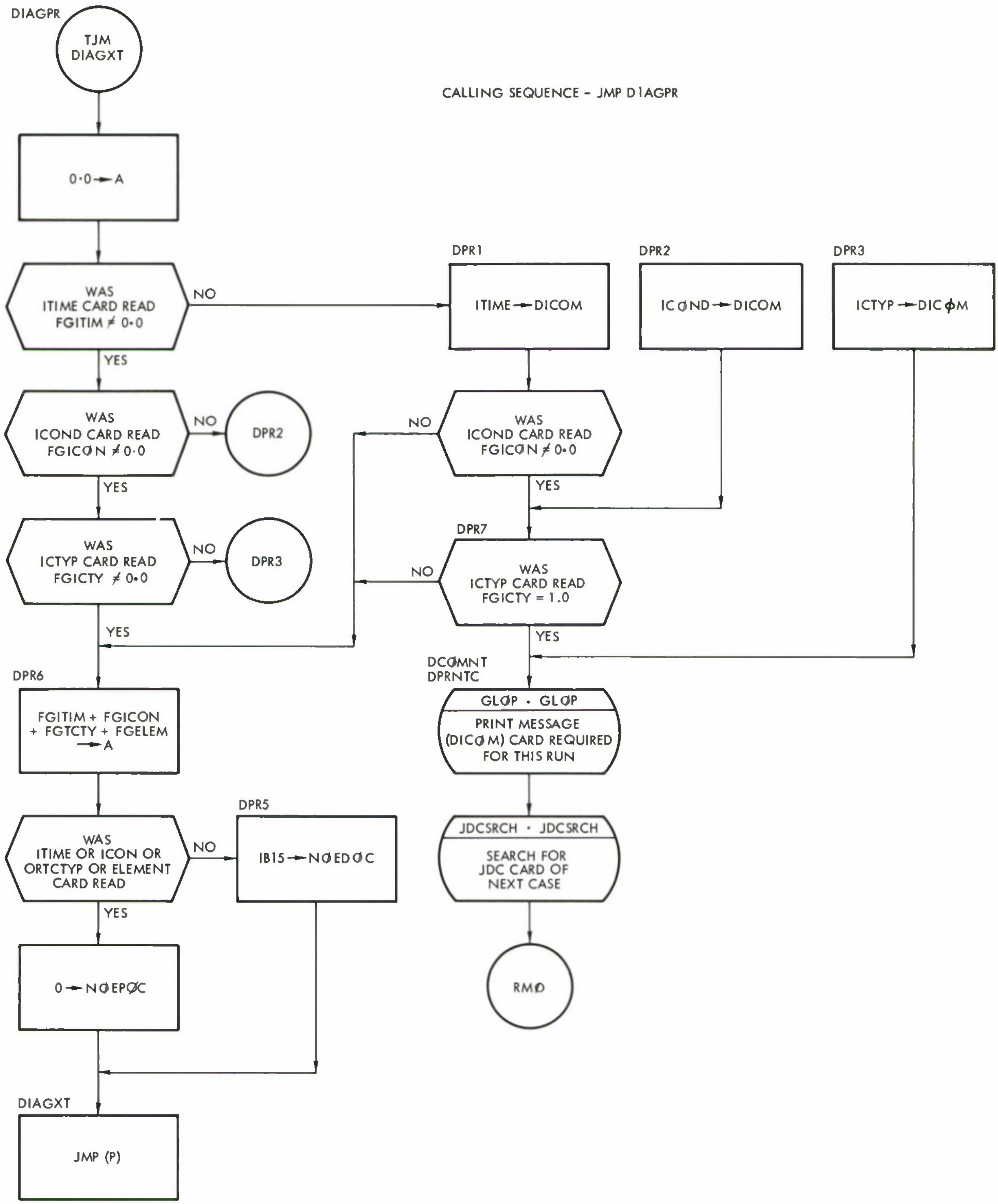


Figure 4-9. DIAGPR Flow Diagram

CALLING SEQUENCE TMD C/HLT, NN; C/HLT, MM
TMQ C/HLT, FRØM; C/HLT, BASE
JMP STØNM

NN IS NUMBER OF SETS TO STORE
MM IS NUMBER OF ITEMS PER SET
FRØM IS LOCATION OF SETS TO BE STORED
BASE IS ORIGIN OF VECTOR IN WHICH ITEMS ARE TO BE STORED
CARD NUMBER (COL 1 AND 2) IS IN CDNUMB

NOTE: INDEX REGISTERS 1 AND 2 ARE NØT SAVED

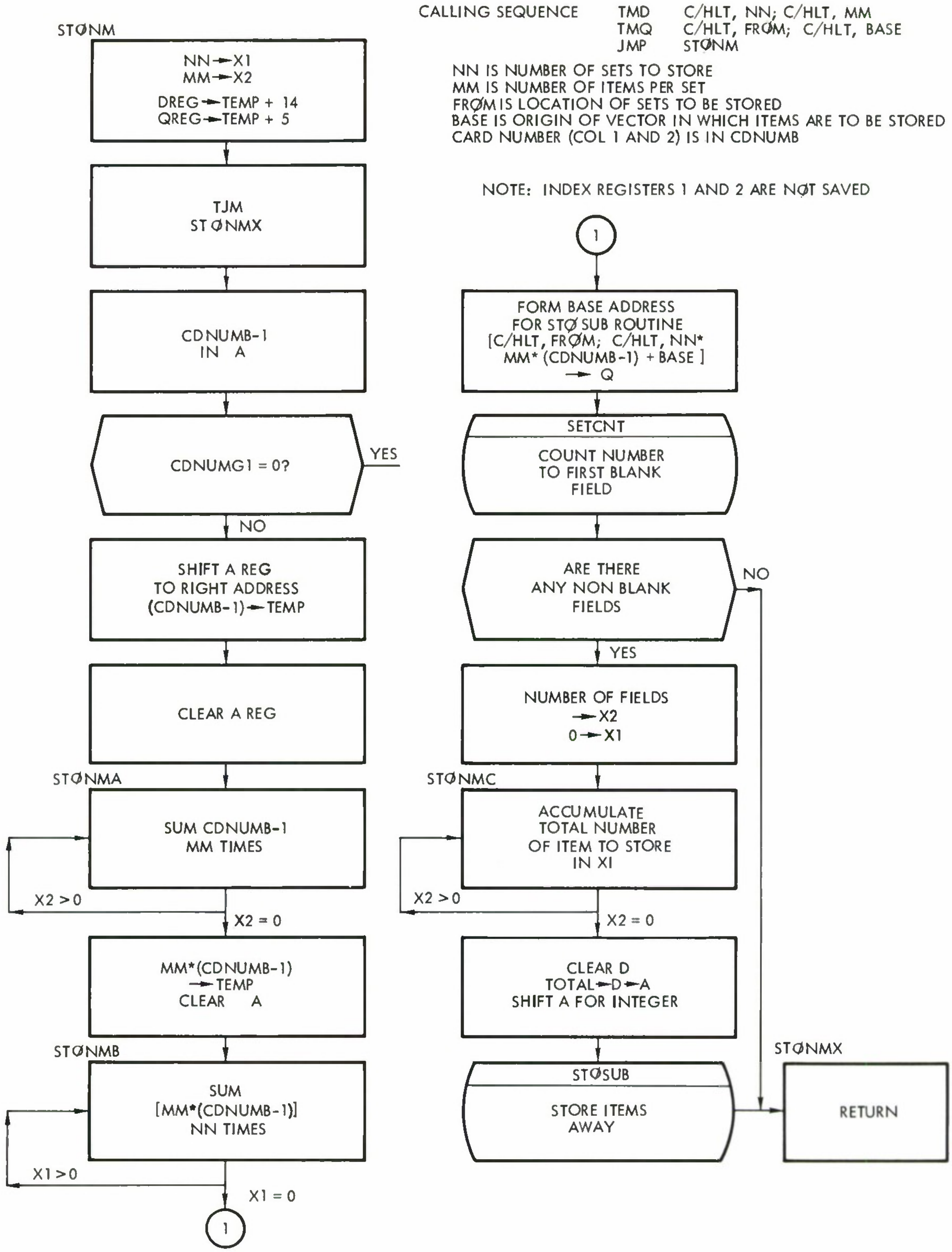


Figure 4-10. STØNM Flow Diagram

CALLING SEQUENCE TMQ C/HLT, FRØM; C/HLT, TØ

TMA N

JMP STØSUB

FRØM IS LOCATION OF SET TO BE STORED

TØ IS LOCATION OF SET TO BE STORED

N IS NUMBER OF ITEMS TO BE STORED AS INTEGER

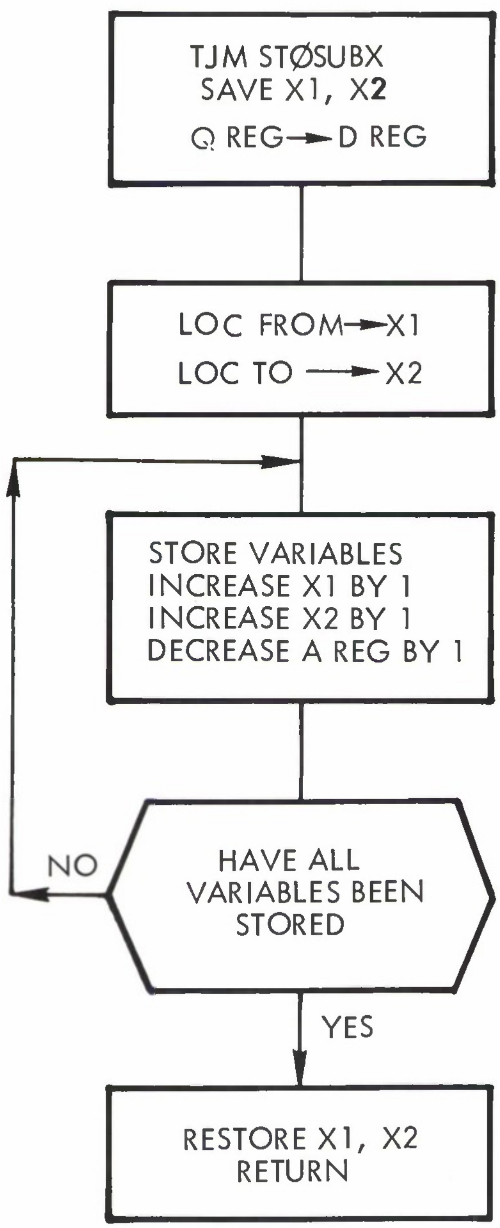
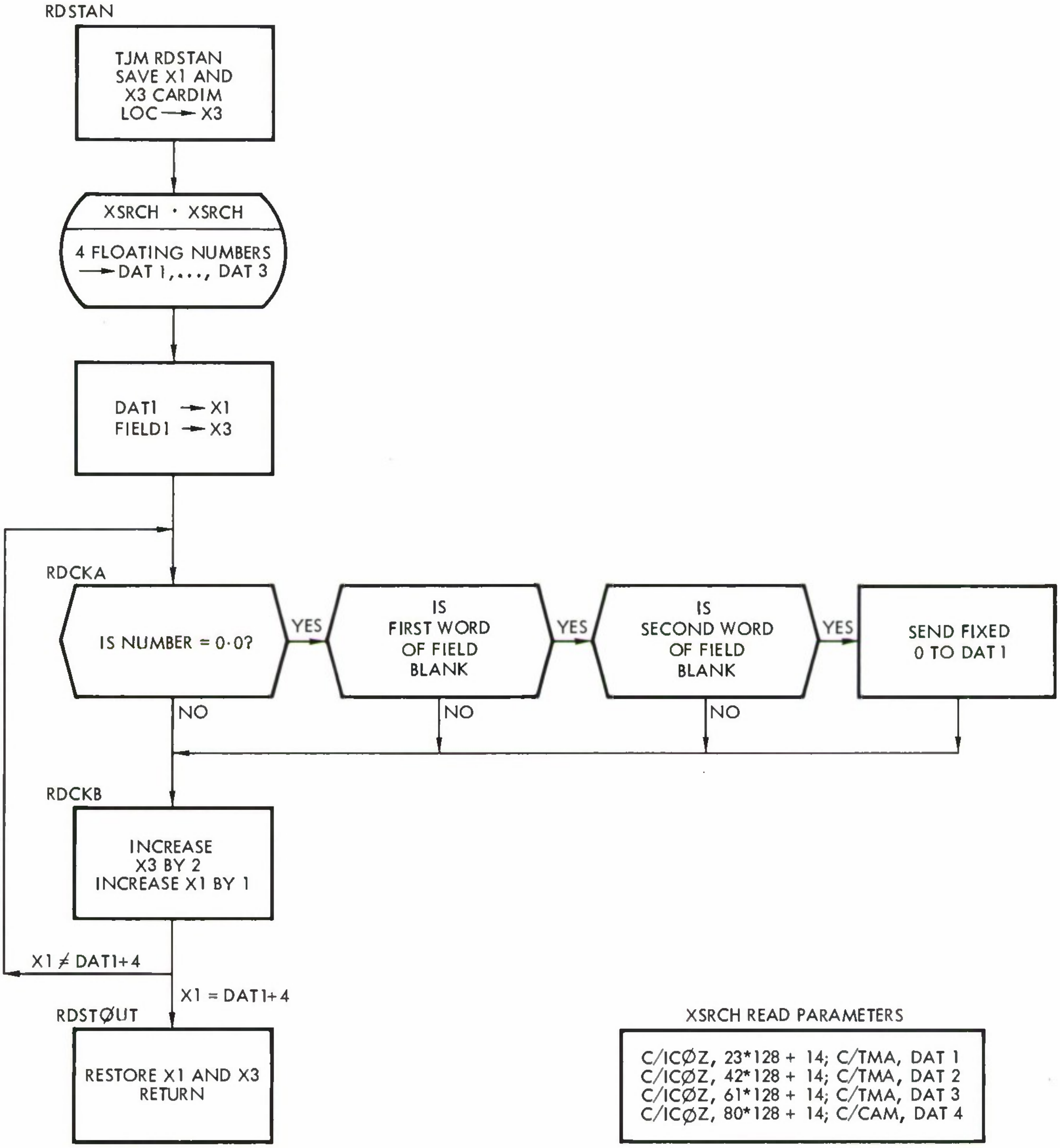


Figure 4-11. STØSUB Flow Diagram



CALLING SEQUENCE JMP RDSTAN

NOTE: IF FIELD BLANK, FIXED ZERO REPLACES FLOATING ZERO

Figure 4-12 RDSTAN Flow Diagram

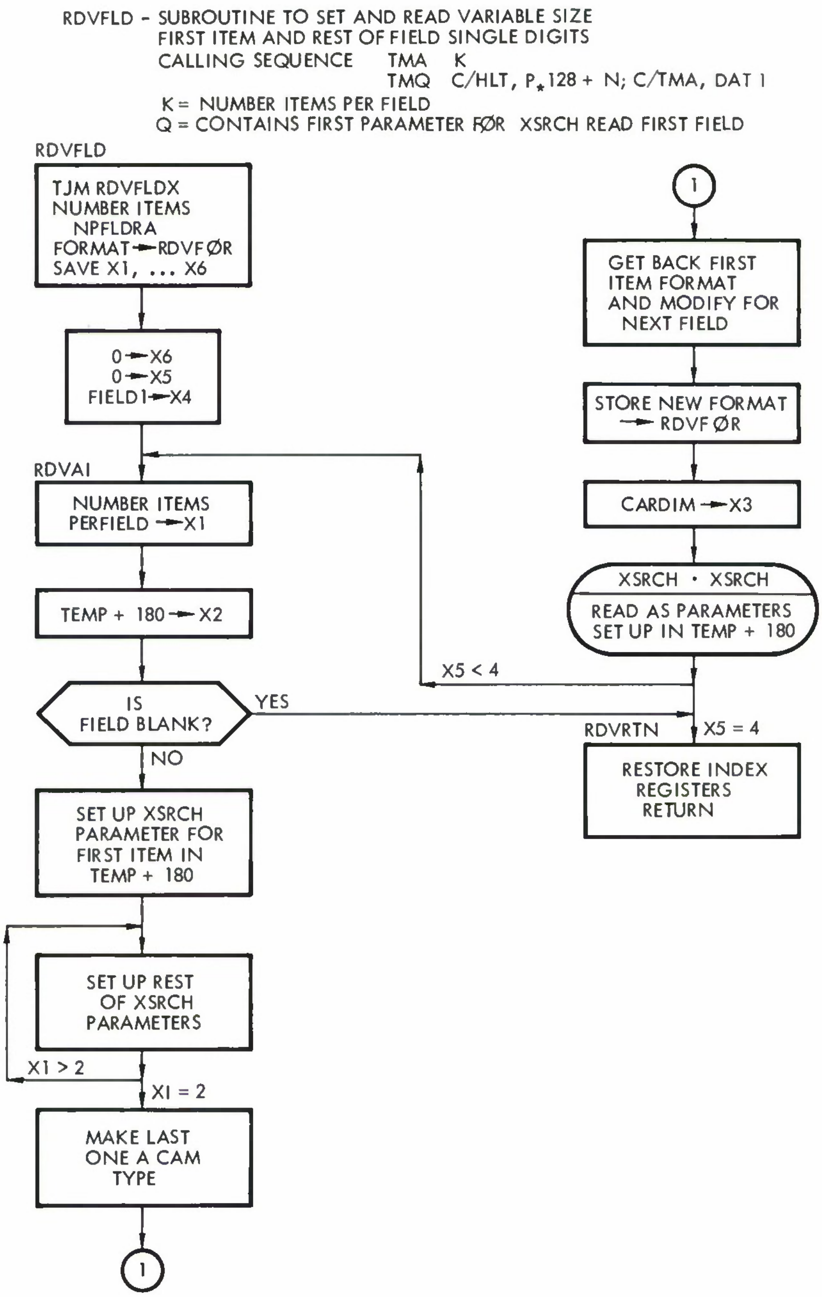


Figure 4-13. RDVFLD Flow Diagram

SETCNT - SUBROUTINE TO COUNT SETS READ BY COUNTING TO FIRST BLANK FIELD
CALLING SEQUENCE JMP SETCNT
NOTE: NUMBER OF SETS IN LEFT ADDRESS OF A REG ON EXIT

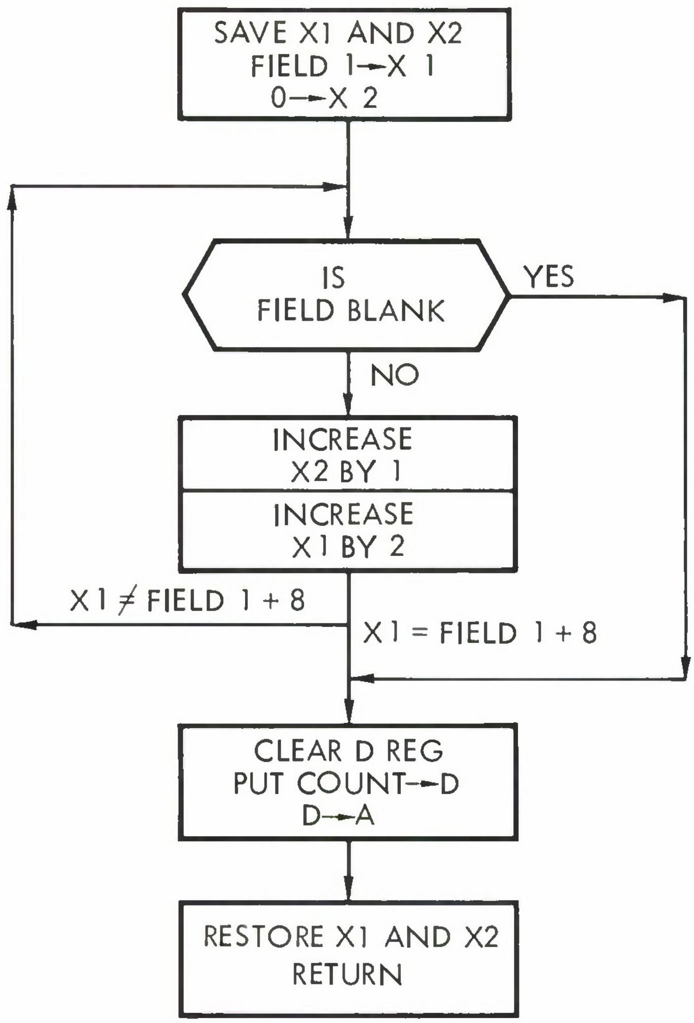


Figure 4-14. SETCNT Flow Diagram

STRDF - SUBROUTINE TO SET UP AND READ FIELDS OF 1 COLUMN ITEMS

CALLING SEQUENCE TMA NUMBER ITEMS PER FIELD (INTEGER)
JMP STRDF

EXIT WITH C/HLT, NUMBER OF SET; C/HLT, NUMBER PER SET IN A

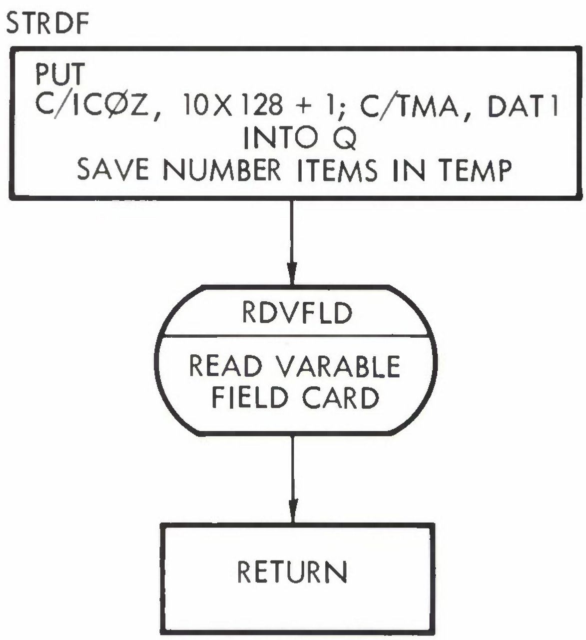


Figure 4-15. STRDF Flow Diagram

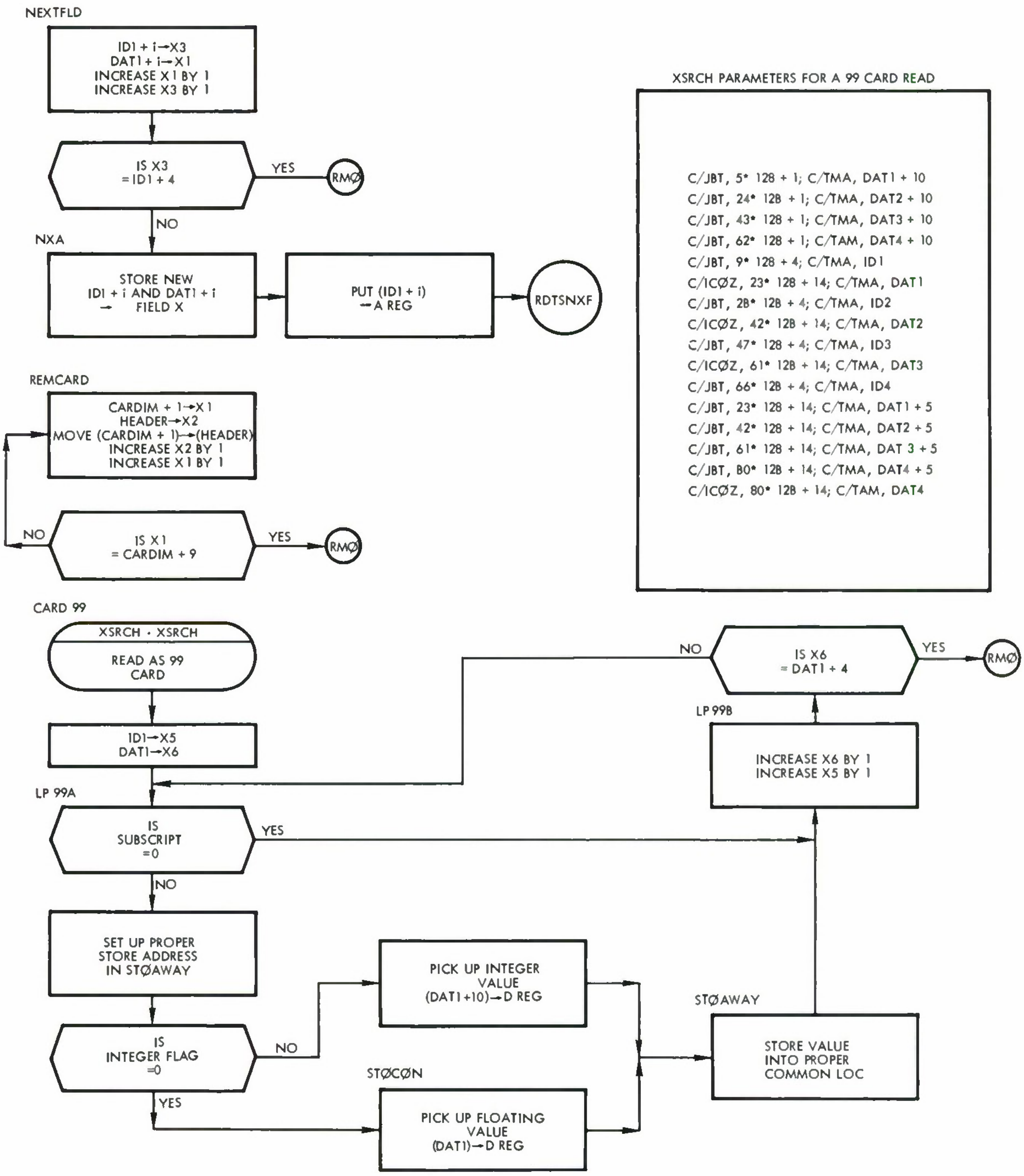


Figure 4-16 a. READPR Flow Diagram (Continued)

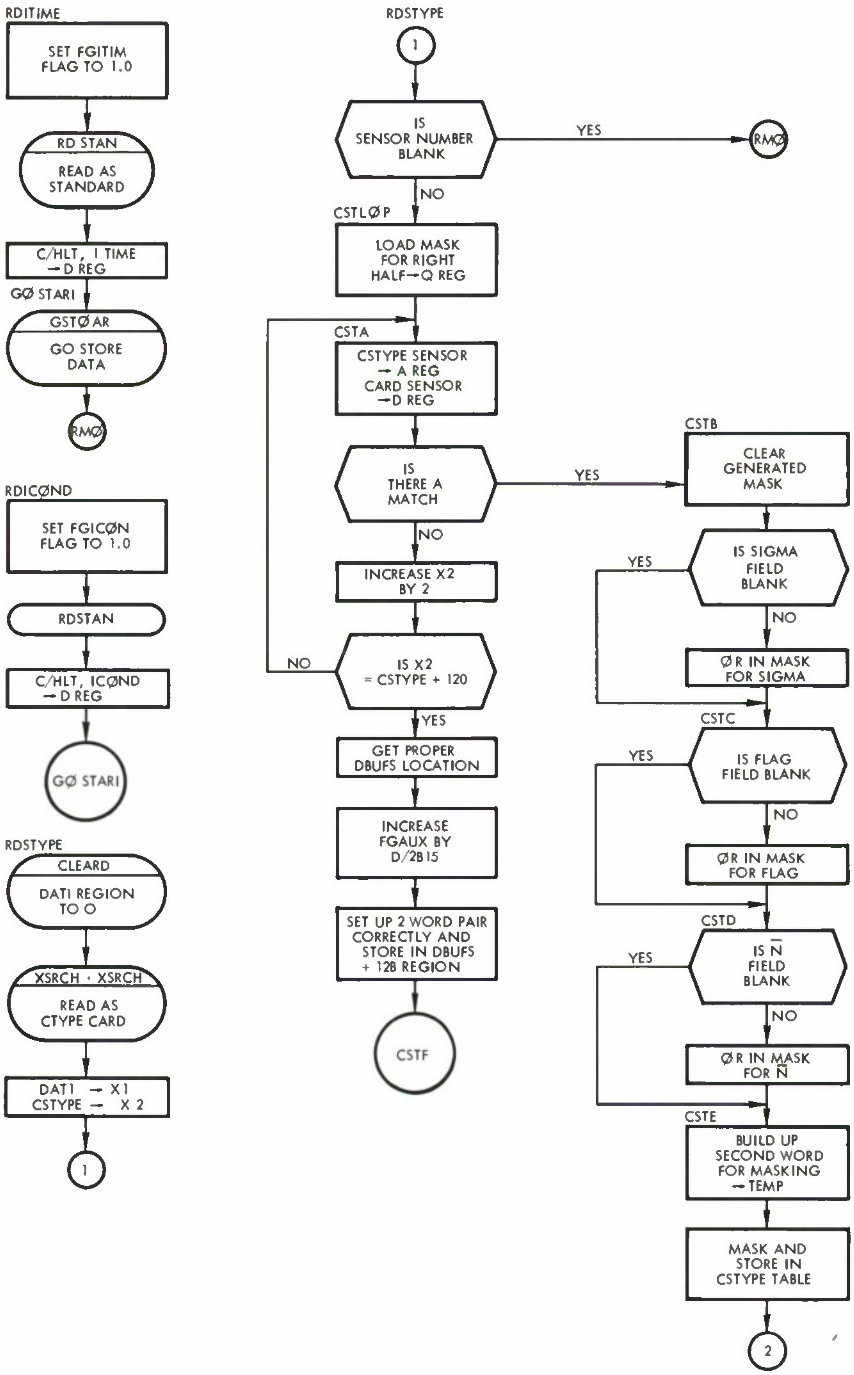


Figure 4-16 b. READPR Flow Diagram (Continued)

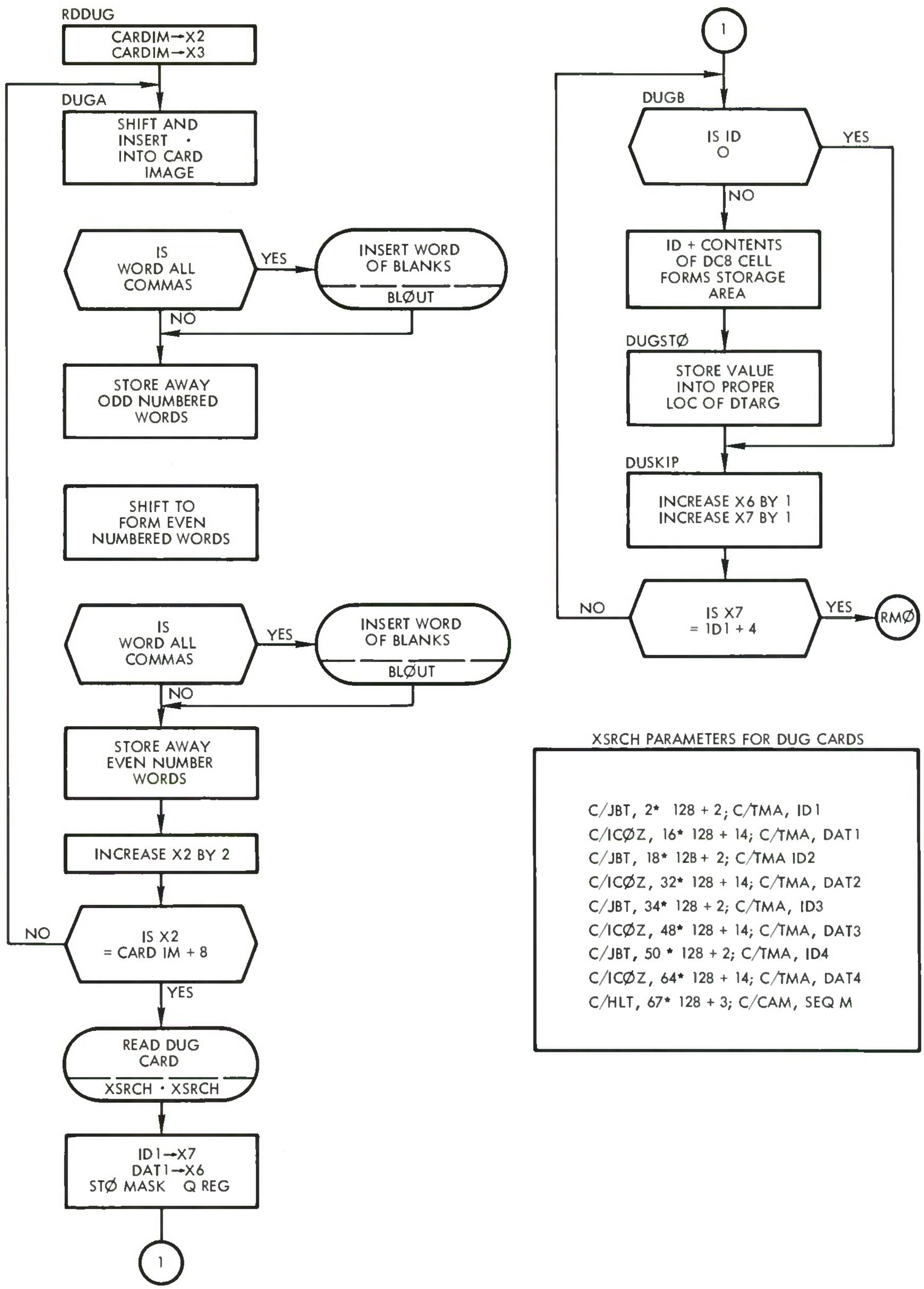
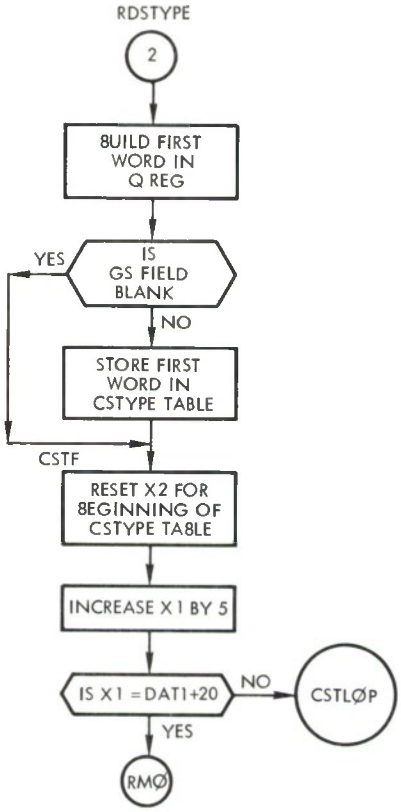


Figure 4-16 c. READPR Flow Diagram (Continued)



XSRCH PARAMETERS FOR STYPE CARD

C/HLT, 13 * 128 + 4; C/TMA, DAT1	SENSOR NO.	FIELD 1
C/HLT, 19 * 128 + 4; C/TMA, DAT1 + 1	G_S	
C/HLT, 15 * 128 + 2; C/TMA, DAT1 + 2	σ_T	
C/HLT, 20 * 128 + 1; C/TMA, DAT1 + 3	FLAG	
C/HLT, 23 * 128 + 3; C/TMA, DAT1 + 4	$\bar{N}_S$	
C/HLT, 32 * 128 + 4; C/TMA, DAT1 + 5	SENSOR NO.	FIELD 2
C/HLT, 38 * 128 + 4; C/TMA, DAT1 + 6	G_S	
C/HLT, 34 * 128 + 2; C/TMA, DAT1 + 7	σ_T	
C/HLT, 39 * 128 + 1; C/TMA, DAT1 + 8	FLAG	
C/HLT, 42 * 128 + 3; C/CAM, DAT1 + 9	$\bar{N}_S$	
C/HLT, 51 * 128 + 4; C/TMA, DAT1 + 10	SENSOR NO.	FIELD 3
C/HLT, 57 * 128 + 4; C/TMA, DAT1 + 11	G_S	
C/HLT, 53 * 128 + 2; C/TMA, DAT1 + 12	σ_T	
C/HLT, 58 * 128 + 1; C/TMA, DAT1 + 13	FLAG	
C/HLT, 61 * 128 + 3; C/TMA, DAT1 + 14	$\bar{N}_S$	
C/HLT, 70 * 128 + 4; C/TMA, DAT1 + 15	SENSOR NO.	FIELD 4
C/HLT, 76 * 128 + 4; C/TMA, DAT1 + 16	G_S	
C/HLT, 72 * 128 + 2; C/TMA, DAT1 + 17	σ_T	
C/HLT, 77 * 128 + 1; C/TMA, DAT1 + 18	FLAG	
C/HLT, 80 * 128 + 3; C/CAM, DAT1 + 19	$\bar{N}_S$	

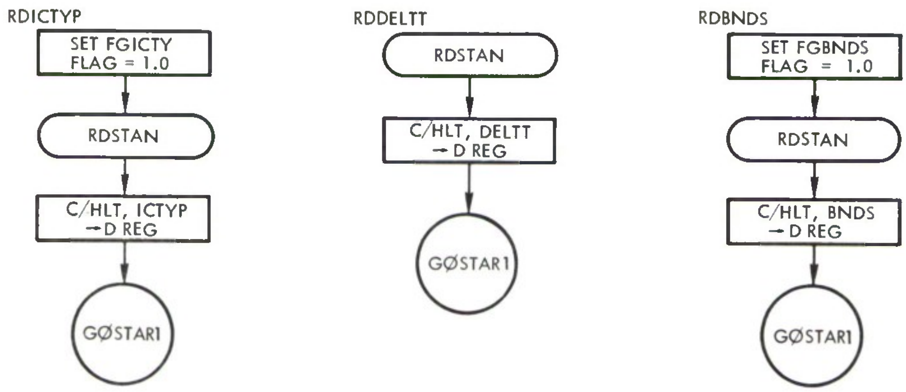


Figure 4-16 d. READPR Flow Diagram (Continued)

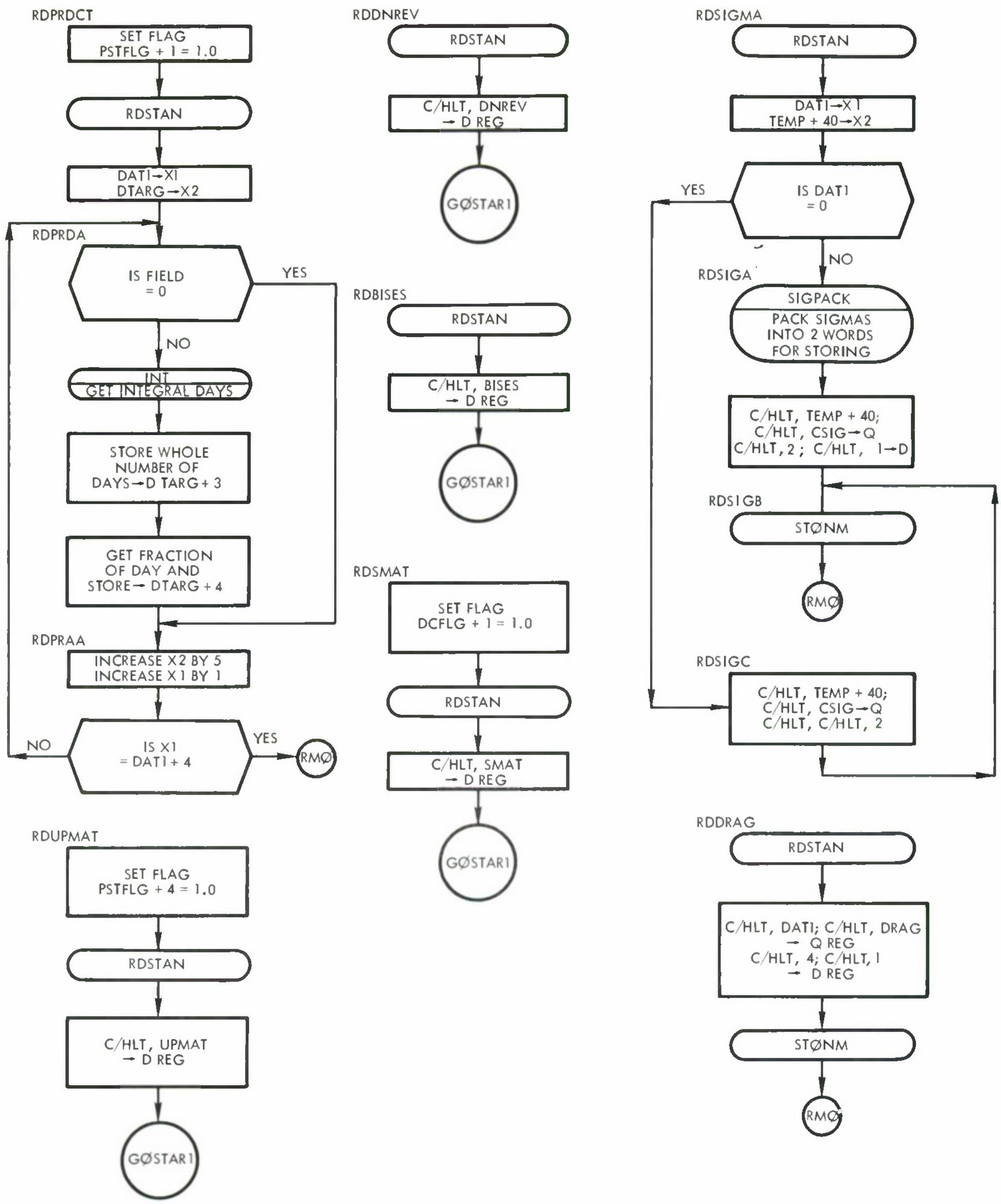


Figure 4-16 e. READPR Flow Diagram (Continued)

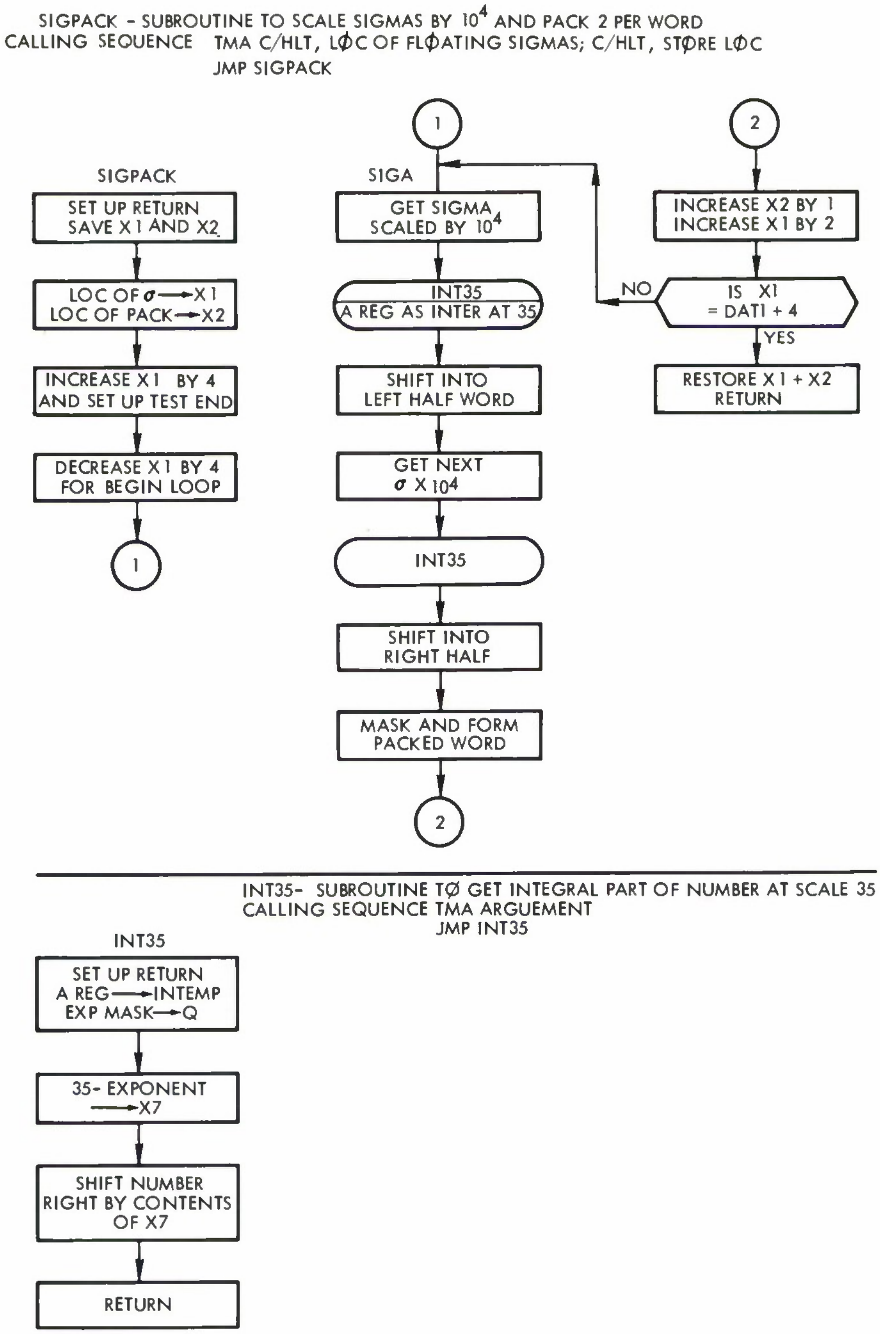


Figure 4-16 f. READPR Flow Diagram (Continued)

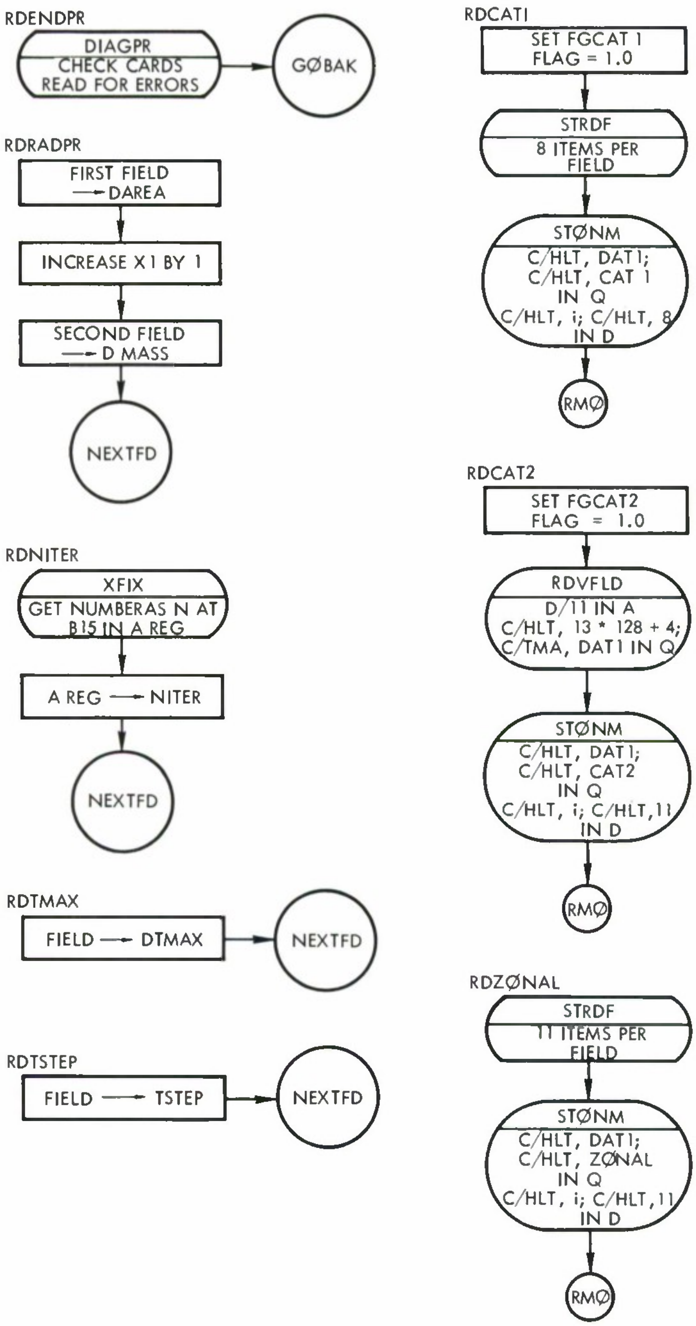
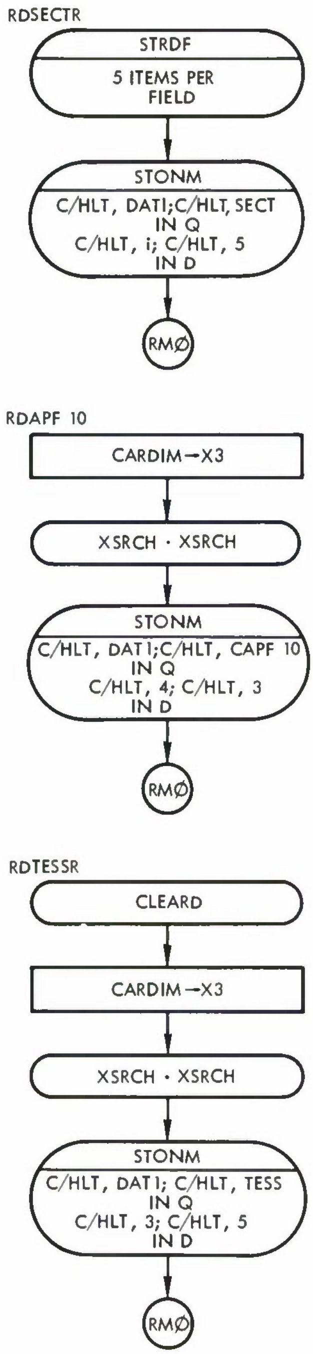


Figure 4-16 g. READPR Flow Diagram (Continued)



XSRCH PARAMETERS FOR APF10 CARD

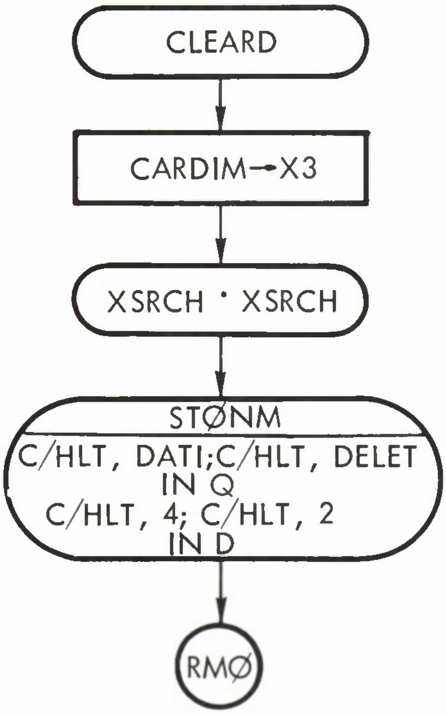
```
C/ICØZ, 13* 128 + 4; C/TMA, DAT1
C/ICØZ, 18* 128 + 5; C/TMA, DAT1 + 1
C/ICØZ, 23* 128 + 5; C/TMA, DAT 1 + 2
C/ICØZ, 32* 128 + 4; C/TMA, DAT 1 + 3
C/ICØZ, 37* 128 + 5; C/TMA, DAT 1 + 4
C/ICØZ, 42* 128 +5; C/TMA, DAT 1 + 5
C/ICØZ, 51* 128 + 4; C/TMA, DAT1 + 6
C/ICØZ, 56* 128 + 5; C/TMA, DAT1 + 7
C/ICØZ, 61* 128 + 5; C/TMA, DAT1 + 8
C/ICØZ, 70* 128 + 4; C/TMA, DAT1 + 9
C/ICØZ, 75* 128 + 5; C/TMA, DAT1 + 10
C/ICØZ, 80* 128 +5; C/CAM DAT1 +11
```

XSRCH PARAMETERS FOR TESSR CARD

```
C/ICØZ, 11* 128 + 2; C/TMA, DAT1
C/ICØZ, 14* 128 + 2; C/TMA, DAT1 + 1
C/ICØZ, 17* 128 + 2; C/TMA, DAT1 + 2
C/ICØZ, 20* 128 + 2; C/TMA, DAT1 +3
C/ICØZ, 23* 128 + 2; C/TMA, DAT1 + 4
C/ICØZ, 30* 128 + 2; C/TMA, DAT 1 + 5
C/ICØZ, 33* 128 + 2; C/TMA, DAT1 + 6
C/ICØZ, 36* 128 + 2; C/TMA, DAT1 + 7
C/ICØZ, 39* 128 +2; C/TMA, DAT1 + 8
C/ICØZ, 42* 128 + 2; C/TMA, DAT1 + 9
C/ICØZ, 49* 128 + 2; C/TMA, DAT1 + 10
C/ICØZ, 52* 128 + 2; C/TMA, DAT1 + 11
C/ICØZ, 55* 128 + 2; C/TMA, DAT1 + 12
C/ICØZ, 58* 128 + 2; C/TMA, DAT1 + 13
```

Figure 4-16 h. READPR Flow Diagram (Continued)

RDDELET



XSRCH PARAMETERS FOR DELET CARD

```
C/ICØZ, 16* 128 + 7; C/TMA, DAT 1
C/ICØZ, 23* 128 + 7; C/TMA, DAT 1 + 1
C/ICØZ, 35* 128 + 7; C/TMA, DAT 1 + 2
C/ICØZ, 42* 128 + 7; C/TMA, DAT 1 + 3
C/ICØZ, 54* 128 + 7; C/TMA, DAT 1 + 4
C/ICØZ, 61* 128 + 7; C/TMA, DAT 1 + 5
C/ICØZ, 73* 128 + 7; C/TMA, DAT 1 + 6
C/ICØZ, 80* 128 + 7; C/CAM, DAT 1 + 7
```

RDJDC

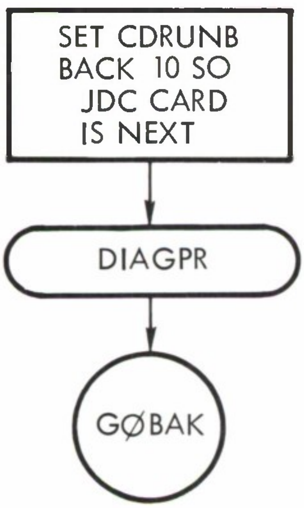


Figure 4-16 i. READPR Flow Diagram (Continued)

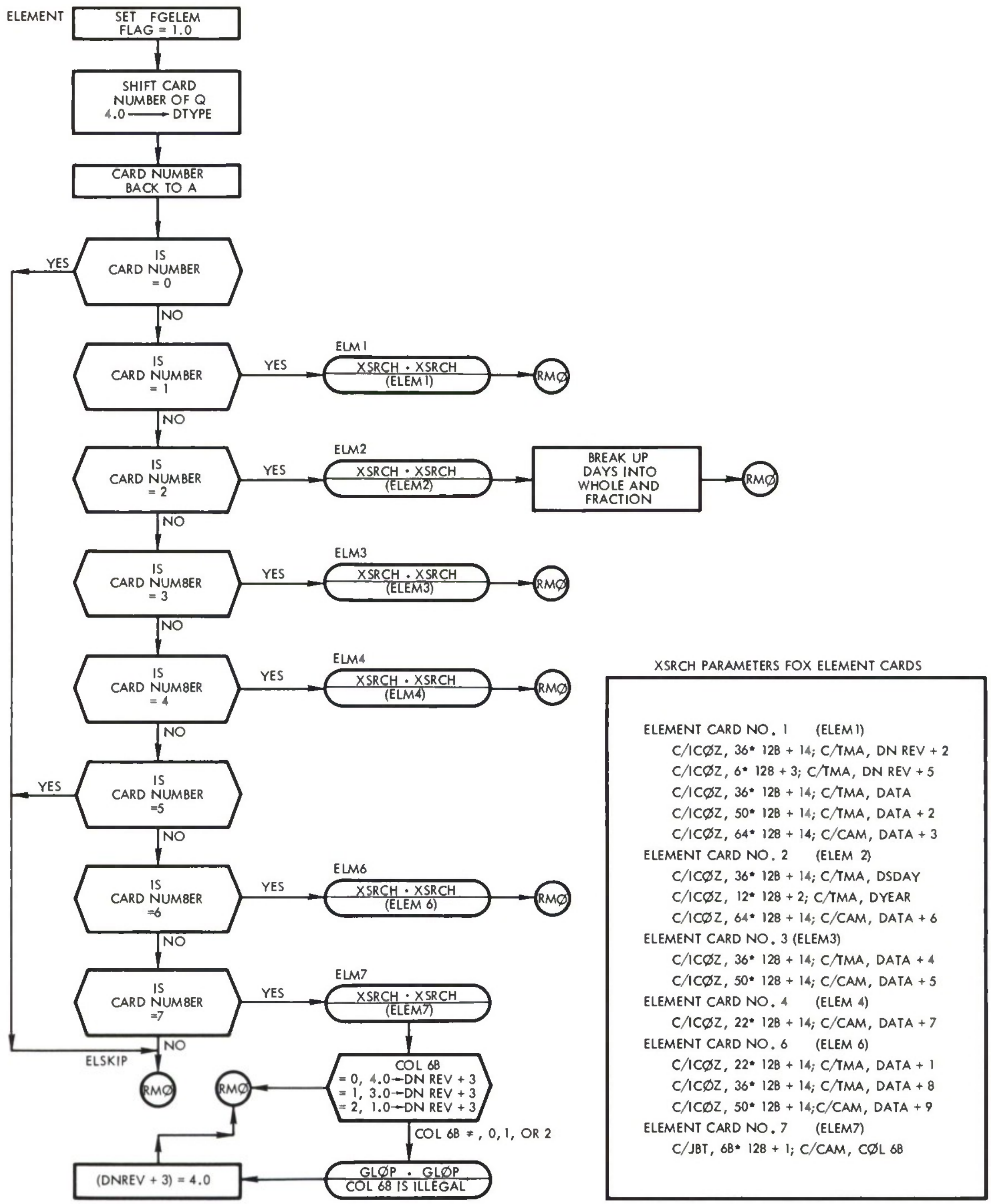


Figure 4-16 j. READPR Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title

RDXYZ

B. Segment

ESPØDEPH

C. Called by subroutine

PØSTPR

FUNCTION

Function is to count, sort, and compute Δt, t sequences from the PRDCT card entries or for the sets of x_T, y_T, z_T, t_D, t_{FD} input.

USAGE

A. Calling sequence

Call RDXYZ

B. Input

1. CØMMØN

DTARG	Array containing the sets of x_T, y_T, z_T, t_D, t_{FD} or the times from the PRDCT card
TEMP	Temporary storage
TEPØCH	Epoch time, minutes from 0^h day of epoch
DSDAY	Epoch day number
DSFDAY	Epoch time, fractions of a day

2. Calling sequence

—

C. Output

1. CØMMØN

DELTT	Δt, t sets to cover the range of times in the DTARG array

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

RDCØM

B. Segment

ESPØD
ESPØDDC
ESPØDEPH

C. Called by subroutines

DPRØS (ESPØD)
ESPØDEPH DRIVER (ESPØDEPH)
INTEG (ESPØDDC)
READPR (ESPØD)

FUNCTION

Function is to read CØMMØN data storage from the work tape into core. This subroutine reads a fixed number of blocks from tape "MT" into consecutive cells, from the start to the end of CØMMØN. This subroutine assumes the next block is a sentinel block on tape and bypasses it. The first block on tape contains blanks except for the first and second words. 70ΔTAPE7 is the first word, and OXXXΔΔΔΔ is the second word. (XXX represents the vehicle number and Δ represents a blank). Subroutine RDCØM checks for correct tape identification. If there is not a match of I.D.'s, a message is printed and the program exits to next case.

USAGE

A. Calling sequence

Call RDCØM

B. Input

1. CØMMØN

CWE	Earth's rotational rate
DBUFS	Auxiliary buffer storage
DVEHN	Vehicle number and name (BCD)
MT	Observation tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

1. Off-Line Comment:

"TAPE ON LOG 7 NOT CORRECT. I.D. IS______"

2. On-Line Comment:

"TAPE ON LOG 7 NOT CORRECT. I.D. IS______"

"TYPE—GO TO RETRY TAPE, - STOP FOR NEXT CASE"

3. Action:

Subroutine ERROR

SUBROUTINES USED

A. Library

GLØP
MXØRD
STØPGØ

B. Program

ERRØR	Error subroutine
FLEX	Flexowriter print routine
REWT	Rewinds observation tape

SUBROUTINE IDENTIFICATION

A. Title
REFRAC

B. Segment
ESPØD

C. Called by subroutine
SWTSN

FUNCTION

This subroutine computes the tropospheric refraction correction for a given elevation angle. The slant range is given to the subroutine for the purpose of computing the altitude of the object at the time the measurement was taken. If the slant range is not available, the altitude is taken to be 70 kilometers.

USAGE

A. Calling sequence
Call REFRAC (R, XNBAR, EZ)

B. Input

1. CØMMØN

CKMER	km/e.r.
CP	4 x 4 array of polynomial coefficients for refraction

2. Calling sequence

R	Slant range (measured), km
XNBAR	$\overline{N}_S$ mean surface value of refractivity
EZ	Measured elevation (mrad)

C. Output

1. CØMMØN

—

2. Calling sequence

EZ	Measured elevation with refraction correction applied (mrad)

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF

SINF

SQRTF

B. Program

ATNQF Arc tangent routine

EQUATIONS

	99 Card Item		99 Card Item
$P_{b_o} = 0.11691966$	608	$Q_{b_o} = 1.0$	612
$P_{b_1} = 0.28291024 \times 10^{-1}$	607	$Q_{b_1} = 0.53601747$	611
$P_{b_2} = 0.49948765 \times 10^{-2}$	606	$Q_{b_2} = 0.62648450 \times 10^{-1}$	610
$P_{b_3} = -0.13841039 \times 10^{-5}$	605	$Q_{b_3} = 0.47029968 \times 10^{-2}$	609
$P_{a_o} = -0.17938996 \times 10^{2}$	616	$Q_{a_o} = 1.0$	620
$P_{a_1} = -0.55482671 \times 10$	615	$Q_{a_1} = 0.77345268$	619
$P_{a_2} = 0.19185634 \times 10^{-2}$	614	$Q_{a_2} = 0.22469128 \times 10^{-1}$	618
$P_{a_3} = 0.82688025 \times 10^{-4}$	613	$Q_{a_3} = 0.24331667 \times 10^{-2}$	617

$$b = \frac{\sum_{i=0}^{3} P_{b_i} E_o^{\,i}}{\sum_{i=0}^{3} Q_{b_i} E_o^{\,i}} \qquad a = \frac{\sum_{i=0}^{3} P_{a_i} E_o^{\,i}}{\sum_{i=0}^{3} Q_{a_i} E_o^{\,i}} \qquad E_o = EZ$$

$$\bar{\tau} = b\bar{N}_s + a \qquad \bar{N}_s = XNBAR$$

$$\zeta = 1 + \frac{h}{R_e} \qquad R_e = \text{radius of the earth, km}$$

$$n_s = 1 + \bar{N}_s \times 10^{-6}$$

$$\tan E_h = \sqrt{\left(\frac{\zeta}{n_s \cos E_o}\right)^2 - 1}$$

$$\bar{\epsilon} = \bar{\tau} - \tan^{-1}\left(\frac{n_s - \cos\bar{\tau} - \sin\bar{\tau}\tan E_o}{n_s \tan E_n + \sin\bar{\tau} - \cos\bar{\tau}\tan E_o}\right)$$

where h, altitude, is computed from radar data, r and E_o (range and elevation).

$$h = \sqrt{R_e^{\,2} + 2R_e r \sin E_o + r^2} - R_e$$

R_e = equatorial radius of Earth, km

r = slant range, km (if slant range is not available, use h = 70 kilometers)

SUBROUTINE IDENTIFICATION

A. Title

REJECT

B. Segment

ESPØDDC

C. Called by subroutines

RADR
INTEG

FUNCTION

Function is to monitor the acceptance or rejection of an observation in the differential correction process. An observation may be rejected if its residual fails to pass either a gross outlier test or fails to pass a K*RMS test where the RMS is computed by observation type using all the observations of the preceding iteration which have passed the gross outlier test (see Figure 4-17). Only the gross outlier test is made on the first iteration. The subroutine has a second entry which when executed computes the RMS by observation type to be used on the next iteration.

USAGE

A. Calling sequence

Call REJECT (I1, I2, I3)

B. Input

1. CØMMØN

PKSUBS	G_s by sensor for gross outlier test
NITCT	Iteration counter
PSIG	Observation weights (σ_R, σ_A, σ_E, σ_{RDT}, σ_{HA}, σ_{DEC})
PRESD	Array containing unweighted residuals (ΔR, ΔA, ΔE, $\Delta \dot{R}$, ΔHA, ΔDEC)
PCSE	cos E
PUI	Array containing the three dimensional vector $U = (u_1, u_2, u_3)$
CKRMS	RMS multiplier for the K*RMS rejection criterion K is nominally set to 1.5

2. Calling sequence

I1 A number 1-6 referring to the type of observation we are testing

I1 = 1 range
2 azimuth
3 elevation
4 range rate
5 hour angle
6 declination

I3 = 1 make residual tests

= 2 compute RMS by observation type for next observation

C. Output

1. CØMMØN

PDELFG Array containing in each cell either

1) word of blanks indicating the observation has been accepted

2) word containing GΔΔΔΔΔΔΔ indicating the observation residual has failed the gross outlier test

3) word containing KΔΔΔΔΔΔΔ indicating the observation residual has failed the K*RMS test

4) word containing *ΔΔΔΔΔΔΔ indicating the observation was deleted by an input DELET card

2. Calling sequence

I2 = 0 residual passed all tests and has been accepted
= 1 residual failed one of the tests and was rejected

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRTF

B. Program

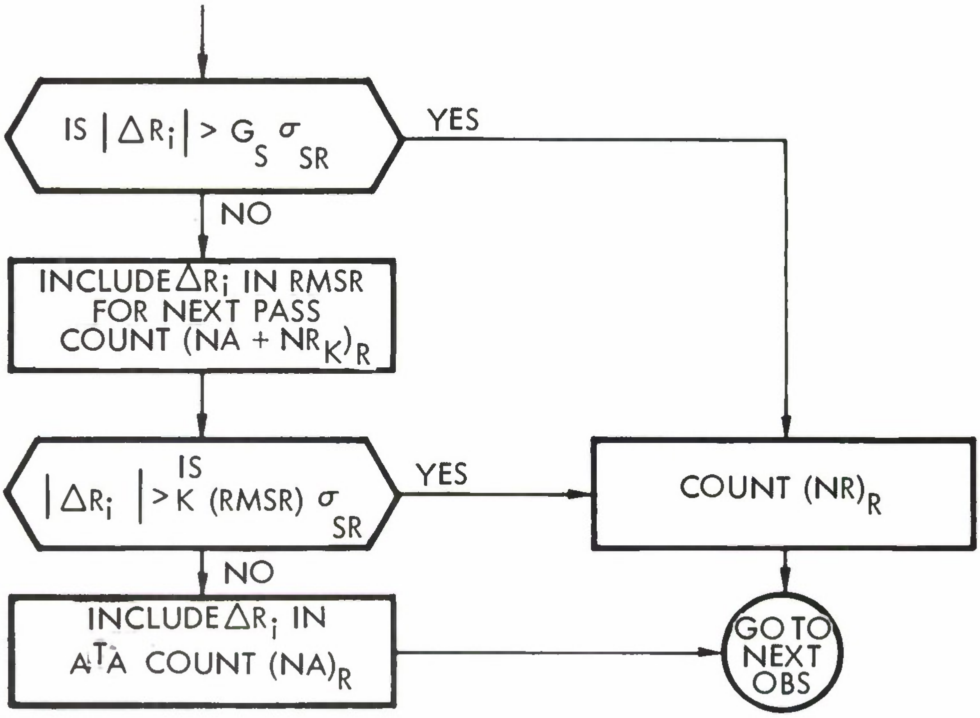
PASTØR Routine to set PDELFG array

METHOD/EQUATIONS

Compute the following for the I1 type observation:

I1	= 1	$\Delta R = \Delta R$
	= 2	$\Delta A = (\cos E)\,\Delta A$
	= 3	$\Delta E = \Delta E$
	= 4	$\Delta \dot{R} = \Delta \dot{R}$
	= 5	$\Delta Ha = (\cos \delta)\,\Delta Ha$
	= 6	$\Delta Dec = \Delta Dec$

For the I1 type observation the following flow diagram gives the sequence in which the rejection testing is carried on.



G_S BY SENSOR

σ_{SR} BY SENSOR AND TYPE

$$RMSR = \sqrt{\frac{\sum \left(\frac{(\Delta R_i)}{\sigma_{SR}} \right)^2}{(NA + NR_K)_R}}$$

Figure 4-17. REJECT Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

REWT

B. Segment

ESPØD

C. Called by subroutines

DPRØS
DRIVER
LØDØBS
RDCØM
WRTCØM

FUNCTION

Function is to rewind the observation tape.

USAGE

A. Calling sequence

Call REWT

B. Input

1. CØMMØN

MT Observation tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

REWT

B. Segment

1. ESPØDDC

2. ESPØDEPH

C. Called by subroutines

INTEG
FIT
WRTCØM

FUNCTION

Function is to rewind the program work tape (MT).

USAGE

A. Calling sequence

Call REWT

B. Input

MT Observation tape number

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

RMAX

B. Segment

ESPØDDC

C. Called by subroutines

PPRINT

FUNCTION

This routine compares the residual output quantities, in temporary storage, with a table of maximum values. If a value exceeds the maximum, it is replaced by the maximum.

USAGE

A. Calling sequence

Call RMAX

B. Input

1. CØMMØN (Table 4-I)

TEMP(1)		Δ range
	(2)	Δ azimuth or Δ right ascension
	(3)	Δ elevation or Δ declination
	(4)	Δ range rate
	(5)	$\Delta \underline{u}$, $\Delta \underline{s}$, Δ station latitude
	(6)	$\Delta \underline{v}$, $\Delta \underline{t}$, or Δ station longitude
	(7)	$\Delta \underline{w}$, $\Delta \underline{w}$, or Δ station height
	(8)	$VM = \sqrt{(\Delta u)^2 + (\Delta v)^2 + (\Delta w)^2}$
	(9)	ΔT = in-plane time differential
	(10)	U = argument of latitude
	(11)	BETA = out-of-plane angle

2. Calling sequence

—

C. Output

1. CØMMØN

TEMP — The output, in TEMP, is in the same format as the input, but if a value exceeds the maximum, it is replaced by the maximum.

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

Table 4-I. Comparison Values

Temp	Compared With	Maximum For
(1)	999.99	Δ Range
(2)	99.99	Δ azimuth or Δ right ascension
(3)	99.99	Δ elevation or Δ declination
(4)	9.999	Δ range rate
(5)	999.99	Δ u, Δ s, or Δ station latitude
(6)	999.99	Δ v, Δ t, or Δ station longitude
(7)	999.99	Δ w, Δ w, or Δ station height
(8)	999.99	VM
(9)	99.999	Δ T
(10)	999.99	U
(11)	9.999	BETA

SUBROUTINE IDENTIFICATION

A. Title

RØTRU

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

INTPL

FUNCTION

This routine rotates a set of vectors from a coordinate system referenced to the mean equator and equinox of 1950.0 to one referenced to the true equator and equinox of date.

USAGE

A. Calling sequence

Call RØTRU (CØØRD, I, NBØD, DATE, NDØ)

B. Input

1. CØMMØN

—

2. Calling sequence

CØØRD	Address of the array containing the vectors to be rotated
I	Location in CØØRD for the X component of the first vector
NBØD	Total number of consecutive vectors to be rotated
DATE	Julian date
NDØ	If 0, a test is made to see if the rotation matrices need be recomputed. If the current date is within .1 days of the previous date, the matrix is not updated. If NDØ $\neq$ 0, the matrices are recomputed at each entrance.

C. Output

1. CØMMØN

—

2. Calling sequence

CØØRD The original vectors now referenced to the true equator and equinox of date

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØS
SIN

B. Program

MULT

EQUATIONS

For a given Julian date rotational matrices $(\bar{a})$ and (a') are computed and the rotation $(a')^T (\bar{a})^T x = x'$ is performed.

$$(a') = \begin{pmatrix} 1 & \psi \cos \epsilon' & \psi \sin \epsilon' \\ -\psi \cos \bar{\epsilon} & 1 & \epsilon' - \bar{\epsilon} \\ -\psi \sin \bar{\epsilon} & \bar{\epsilon} - \epsilon' & 1 \end{pmatrix}$$

ϵ', ψ, $\bar{\epsilon}$ are computed as follows:

$$\text{Set } D = \text{Julian date} - 2433282.5$$

$$T = d/2433282.5$$

$$\psi = d\psi + \Delta\psi$$

The coefficients a_i, b_i, c_i, e_i used below are listed in Table 4-II.

$$d\psi = e_1 \sin 2\leftmoon + e_2 \sin (\leftmoon - \Gamma') + e_3 \sin 2 (\leftmoon - \Gamma') + e_4 \sin (2\leftmoon - \Omega)$$

$$+ e_5 \sin (3\leftmoon - \Gamma') + e_6 \sin (\leftmoon + \Gamma' - 2) + e_7 \sin (\leftmoon + \Gamma')$$

$$+ e_8 \sin 2 (\leftmoon - L) + e_9 \sin (\leftmoon - \Gamma'\Omega) + e_{10} \sin (\leftmoon - \Gamma' - \Omega)$$

$$+ e_{11} \sin (3\leftmoon - 2L + \Gamma') + e_{12} \sin (3\leftmoon - \Gamma' - \Omega)$$

$$\Delta\psi = -\sin \Omega \, (c_1 + c_2 T) + c_3 \sin 2\Omega + c_4 \sin 2L + c_5 \sin (L - \Gamma)$$

$$+ c_6 \sin (3L - \Gamma) + c_7 \sin (L + \Gamma) + c_8 \sin (2L - \Omega) + c_9 \sin (2\Gamma' - \Omega)$$

$$+ c_{10} \sin 2 (L - \Gamma')$$

$$\epsilon' = \bar{\epsilon} + \Delta\epsilon + d\epsilon$$

$\bar{\epsilon} = P_1(T)$, where P_1 is a cubic in. T (see Table 4-III)

$$d\epsilon = b_1 \cos 2\leftmoon + b_2 \cos (2\leftmoon - \Omega) + b_3 \cos (3\leftmoon - \Gamma') + b_4 \cos (\leftmoon + \Gamma')$$

$$+ b_5 \cos (\leftmoon - \Gamma' + \Omega) + b_6 \cos (\leftmoon - \Gamma' - \Omega) + b_8 \cos (3\leftmoon - 2L + \Gamma')$$

$$+ b_9 \cos (3\leftmoon - \Gamma' - \Omega)$$

$$\Delta\epsilon = a_1 \cos \Omega + a_2 \cos 2\Omega + a_3 \cos 2 + a_4 \cos (3L - \Gamma) + a_5 \cos (L + \Gamma)$$

$$+ a_6 \cos (2L - \Omega) + a_7 \cos (2\Gamma' - \Omega)$$

$$\left.\begin{aligned}
\Omega &= P_2(T) - 9.24220286 \times 10^{-4}\, D \\
\mathbb{C} &= P_9(T) + 0.229971498\, D \\
\Gamma' &= P_3(T) + 1.94436796 \times 10^{-3}\, D \\
L &= P_4(T) + 1.72027908 \times 10^{-2}\, D \\
\Gamma &= P_5(T) + 8.21498543 \times 10^{-7}\, D
\end{aligned}\right\}\quad \text{See Table 4-III for polynomials}$$

$$(\bar{a}) = \begin{bmatrix}
-\sin \zeta_o \sin z + \cos \zeta_o \cos z \cos \theta & \sin \zeta_o \cos z + \cos \zeta_o \sin z \cos \theta & \cos \zeta_o \sin \theta \\
-\cos \zeta_o \sin z - \sin \zeta_o \cos z \cos \theta & \cos \zeta_o \cos z - \sin \zeta_o \sin z \cos \theta & -\sin \zeta_o \sin \theta \\
-\cos z \sin \theta & -\sin z \sin \theta & \cos \theta
\end{bmatrix}$$

$$\left.\begin{aligned}
\zeta_o &= P_6(T) \\
z &= P_7(T) \\
\theta &= P_8(T)
\end{aligned}\right\}\quad \text{See Table 4-III for polynomials}$$

Table 4-II. List of Coefficients a_i, b_i, c_i, and e_i

$a = 0.446512811 \times 10^{-4}$

$a_2 = -0.4363323 \times 10^{-6}$

$a_3 = 0.26713223 \times 10^{-5}$

$a_4 = 0.106658809 \times 10^{-6}$

$a_5 = -0.436332308 \times 10^{-7}$

$a_6 = -0.33936879 \times 10^{-7}$

$a_7 = -0.145438284 \times 10^{-7}$

$c_1 = 0.835936092 \times 10^{-4}$

$c_2 = 0.824179359 \times 10^{-7}$

$c_3 = 0.101325087 \times 10^{-5}$

$c_4 = -0.616677172 \times 10^{-5}$

$c_5 = 0.610865228 \times 10^{-6}$

$c_6 = -0.242391318 \times 10^{-6}$

$c_7 = 0.101810284 \times 10^{-6}$

$c_8 = 0.581718229 \times 10^{-7}$

$c_9 = 0.242391318 \times 10^{-7}$

$c_{10} = 0.193923526 \times 10^{-7}$

$b_1 = 0.426628269 \times 10^{-6}$

$b_2 = 0.872664616 \times 10^{-7}$

$b_3 = 0.533285342 \times 10^{-7}$

$b_4 = -0.242391318 \times 10^{-7}$

$b_5 = -0.145438284 \times 10^{-7}$

$b_6 = 0.145438284 \times 10^{-7}$

$b_7 = 0.969530374 \times 10^{-8}$

$b_8 = 0.969530374 \times 10^{-8}$

$e_1 = -0.989008255 \times 10^{-6}$

$e_2 = 0.329657778 \times 10^{-6}$

$e_3 = 0.145443521 \times 10^{-7}$

$e_4 = -0.16483587 \times 10^{-6}$

$e_5 = -0.126047671 \times 10^{-6}$

$e_6 = 0.727208875 \times 10^{-7}$

$e_7 = 0.533285342 \times 10^{-7}$

$e_8 = 0.290876567 \times 10^{-7}$

$e_9 = 0.290876567 \times 10^{-7}$

$e_{10} = 0.290876567 \times 10^{-7}$

$e_{11} = 0.242391318 \times 10^{-7}$

$e_{12} = -0.193906069 \times 10^{-7}$

RØTRU

Table 4-III. List of Polynomials

	1	T	T^2	T^3
P	0.409206174	$-2.27132954 \times 10^{-4}$	$-1.54548898 \times 10^{-8}$	$8.77900593 \times 10^{-9}$
P_2	0.211408064	$3.62941205 \times 10^{-5}$	3.6320311×10^{-5}	3.4906584×10^{-8}
P_3	3.64501514	$-1.80362321 \times 10^{-4}$	$-1.80519402 \times 10^{-4}$	$-2.09439501 \times 10^{-7}$
P_4	4.88833919	$5.27089417 \times 10^{-6}$	$5.27089417 \times 10^{-6}$	0
P_5	4.92323384	$7.94561125 \times 10^{-6}$	$7.98488118 \times 10^{-6}$	$5.23598768 \times 10^{-8}$
P_6	0	$1.11749403 \times 10^{-2}$	$1.46413572 \times 10^{-6}$	$8.67777698 \times 10^{-8}$
P_7	0	$1.11749403 \times 10^{-2}$	$5.29901341 \times 10^{-6}$	$9.30784076 \times 10^{-8}$
P_8	0	$9.71711062 \times 10^{-3}$	$-2.06530564 \times 10^{-6}$	$-2.01672785 \times 10^{-7}$
P_9	1.1235635	$-1.97497085 \times 10^{-5}$	$-1.97248377 \times 10^{-5}$	3.3161255×10^{-8}

SUBROUTINE IDENTIFICATION

A. Title

RPRESS

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

DAUX

FUNCTION

Function is to compute the perturbative acceleration on a spacecraft due to solar radiation pressure.

USAGE

A. Calling sequence

Call RPRESS

B. Input

1. CØMMØN

CGMR	Array containing ratios of moon, sun, Venus, Mars, Saturn and Jupiter GM to that of the Earth
TLIST	Numerical integration working storage
DBASE	Days from 1950.0 to epoch
CMU	GM of the Earth
CERAU	Earth radii per astronomical unit, conversion factor
TALFA	A constant used in the simulation of radiation pressure
CLIGHT	The speed of light

2. Calling sequence

—

C. Output

1. CØMMØN

TRPRES — Three-cell array containing the acceleration due to radiation pressure in the x, y, and z directions

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

SQRT

B. Program

DØT
INTPL
MAGN

EQUATIONS

A. Read the ephemeris tape for the position of the sun, and compute the velocity of the sun analytically:

$$V_s = \left[\mu\left(\frac{2}{R_s} - \frac{1}{R_{au}}\right)\right]^{1/2}$$

where

$$R_s = \left(x_s^2 + y_s^2 + z_s^2\right)^{1/2}$$

$$R_{au} = 1 \text{ a.u.}$$

$$\mu = \text{sum of Earth GM and Sun GM}$$

$$\begin{bmatrix} \dot{x}_s' \\ \dot{y}_s' \\ \dot{z}_s' \end{bmatrix} = \begin{bmatrix} 0 \\ -\sin(23.5^\circ) \\ \cos(23.5^\circ) \end{bmatrix} \times \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix}$$

$$\dot{x}_s = \frac{V_s}{R} \cdot \dot{x}_s'$$

$$\dot{y}_s = \frac{V_s}{R} \cdot \dot{y}_s'$$

$$\dot{z}_s = \frac{V_s}{R} \cdot \dot{z}_s'$$

B. Compute the position and velocity of the vehicle referenced to the sun.

$$x_{vs} = x - x_s \qquad \dot{x}_{vs} = \dot{x} - \dot{x}_s$$

$$y_{vs} = y - y_s \qquad \dot{y}_{vs} = \dot{y} - \dot{y}_s$$

$$z_{vs} = z - z_s \qquad \dot{z}_{vs} = \dot{z} - \dot{z}_s$$

C. Acceleration due to solar radiation pressure:

$$u = (\bar{x}_{vs} \cdot \dot{x}_{vs}) \qquad \text{if } \bar{x}_{vs} = \frac{\vec{x}_{vs}}{|R_{vs}|}$$

$$\ddot{x} = ac\left(1 - \frac{u}{c}\right)\frac{x_{vs}}{R_{vs}} - a\,\frac{\dot{x}_{vs}}{R_{vs}^2}$$

$$\ddot{y} = ac\left(1 - \frac{u}{c}\right)\frac{y_{vs}}{R_{vs}} - a\,\frac{\dot{y}_{vs}}{R_{vs}^2}$$

$$\ddot{z} = ac\left(1 - \frac{u}{c}\right)\frac{z_{vs}}{R_{vs}} - a\,\frac{\dot{z}_{vs}}{R_{vs}^2}$$

where

$$C = \text{speed of light}$$

$$R_{vs} = \left(x_{vs}^2 + y_{vs}^2 + z_{vs}^2\right)^{1/2}$$

$$a = \frac{S}{C^2} R^2 \frac{A}{m} \text{ from TPRLM subroutine}$$

SUBROUTINE IDENTIFICATION

A. Title

SDELET

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

The function is to move observation deletion numbers from DATA storage starting at DATA (557) to IVSTR variable storage starting at IVSTR (NIDLED).

USAGE

A. Calling sequence

Call SDELET

B. Input

1. CØMMØN

DATA	Input storage
NIDENT	Number of entries in the NIDLED list
NIDLED	Identifies the starting location of where the observation deletion table begins

2. Calling sequence

—

C. Output

1. CØMMØN

IVSTR (NIDLED)	Array containing pairs of residual numbers for deletion purposes

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SELECT

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutines

INTEG (ESPØDDC)
PØSTPR (ESPØDEPH)

FUNCTION

This subroutine selects the next time whether it be an observation time (ESPØDDC) or a straight prediction time (ESPØDEPH) to which the numerical integration is to be carried.

USAGE

A. Calling sequence

Call SELECT

B. Input

1. CØMMØN

IPFRST	0 to indicate first time in RADR
PUBS	Sensor number, time, R, A, E, $\dot{R}$, α, δ table
TEPØCH	Epoch time, minutes from midnight
TLIST	Numerical integration working storage
TMINUS	Flag to indicate integration times before epoch
TUBSEF	EØF flag for reading observations

2. Calling sequence

—

C. Output

1. CØMMØN

TG	Time, minutes from 0^h of epoch day, to integrate

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

SETIC	Initialize the integration list
UBSGET	Gets next observation time from variable storage

METHOD

See Figure 4-18.

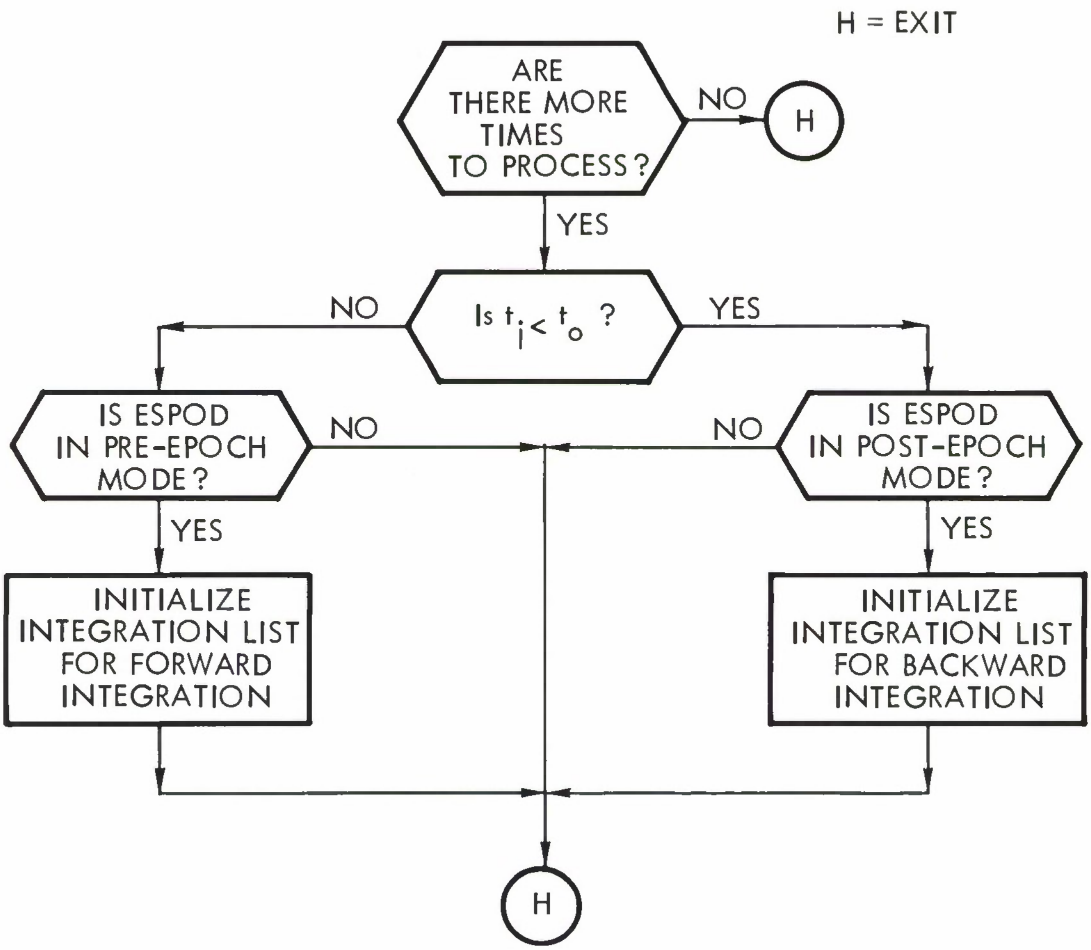


Figure 4-18. Differential Corrector Flow Diagram

SUBROUTINE IDENTIFICATION

A. Title

SENIN

B. Segment

ESPØD

C. Called by subroutine

LØDSEN

FUNCTION

This subroutine inputs sensor location cards, computes number of sensors and sets up master sensor table with correct units and values.

USAGE

A. Calling sequence

Call SENIN

B. Input

1. CØMMØN

CAE	a_e
CBE	b_e
CDEG	Degrees/radian
CMTER	Meters/e.r.
DTMP	Saves station number and code word for those stations with code word $\neq 0$
NDPR	Number of all differential + initial parameters to solve for (Category 1)
NPR	Number of all parameters to solve for
NSTAT	Identifies the starting location of the master sensor table
NUBS	Identifies the starting location of the observation table
TALFAG	α_{go} for midnight day of epoch
TEMP(1)	Sensor number (BCD)
TEMP(2)	Latitude, degrees
TEMP(3)	Longitude, degrees
TEMP(4)	Altitude, meters

TEMP(6) TEMP(7)	Sensor name
TEMP(9)	Classified or unclassified flag

2. Calling sequence

—

C. Output

1. CØMMØN

VSTR(NSTAT)	Sensor number
(NSTAT+1)	Latitude, radians
(NSTAT+2)	Longitude, radians
(NSTAT+3)	Altitude, e.r.
(NSTAT+4)	cos ϕ*
(NSTAT+5)	Sin ϕ*
(NSTAT+6)	$a_{go} + \lambda$, radians
(NSTAT+7)	W_1^S, e.r.
(NSTAT+8)	W_3^S, e.r.
(NSTAT+9)	Code word
(NSTAT+10)	0.0
(NSTAT+11)	0.0
(NSTAT+12)	0.0
(NSTAT+13)	0.0
(NSTAT+14)	0.0

2. Calling sequence

—

D. Error/action messages

SUBROUTINES USED

A. Library

CØSF
SINF
SQRTF

B. Program

PIMØD	Takes principal value of angle between 0 and 2π

EQUATIONS

$$a g_o + \lambda$$

$$W_1^S = (a_e A_s + h) \cos \phi^*$$

$$W_3^S = (b_e A_s + h) \sin \phi^*$$

where

$$A_s = (\cos^2 \phi^* + b_e^2 \sin^2 \phi^*)^{-1/2}$$

$$B_s = \left(\sin^2 \phi^* + \frac{1}{b_e^2} \cos^2 \phi^*\right)^{-1/2}$$

a_e found in CØMMØN in CAE

b_e found in CØMMØN in CBE

SUBROUTINE IDENTIFICATION

A. Title

SENRD

B. Segment

ESPØD

C. Called by subroutine

LØDSEN

FUNCTION

Function is to read one sensor card in SPADATS format, from the input tape.

USAGE

A. Calling sequence

Call SENRD (SEØF)

B. Input

1. CØMMØN

—

2. Calling sequence

SEØF = -1.	(if more sensors)
SEØF = 1.	(ENDSN card detected)

C. Output

1. CØMMØN

TEMP(1)	Sensor number
TEMP(2)	ϕ (latitude, degrees)
TEMP(3)	λ (longitude, degrees)
TEMP(4)	h (altitude, meters)
TEMP(6) TEMP(7)	Sensor name
TEMP(9)	Classified or unclassified flag

2. Calling sequence

—

D. Error/action messages

No message but a jump to "ERRØR" if the "A" register is zero when returning from XSRCH

SUBROUTINES USED

A. Library

—

B. Program

ERRØR	Error routine
IDSUB	Strips blanks from I.D.
READPR.RDØNE	Reads preliminary data
XSRCH	Card image scan and convert

SUBROUTINE IDENTIFICATION

A. Title

SENSCH

B. Segment

ESPØD

C. Called by subroutine

LØDSEN

FUNCTION

This subroutines searches the required sensor table, located at DBUFS(129) through DBUFS(256), for a match with sensor card I.D. If a match is found, DBUFS(I) is set = 0. If no match is found then SNAME is set = 0.

USAGE

A. Calling sequence

Call SENSCH(SNAME)

B. Input

1. CØMMØN

DBUFS	Temporary buffer storage

2. Calling sequence

SNAME	Flag for sensor card I.D.

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SETCØN

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

This subroutine stores values from the B2 master assign deck constants into the ESPØD constants pool. It also computes several other constants used by ESPØD whose values are dependent on the constants picked out of the B2 constants pool.

USAGE

A. Calling sequence

Call SETCØN

B. Input

1. CØMMØN

—

2. Calling sequence

—

C. Output

1. CØMMØN

CBE	b_e
CDTER	km/e. r.
CELLIP	Ellipticity of the Earth
CFTER	ft/ e. r.
CKMER	km/e. r.
CLIGHT	Speed of light
CMTER	Meters/e. r.
CNMER	Nautical miles/e. r.
CVTERM	Convert e. r. /min to km/sec
CWE	Earth's rotational rate
CGMR(2)	Earth to moon mass ratio

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SETIC

B. Segment

ESPØDDC

C. Called by subroutines

INTEG (ESPØDDC)
SELECT (ESPØDDC)

FUNCTION

The function is to initialize the integration list and other parameters which must be re-initialized each time the integration is re-started.

USAGE

A. Calling sequence

Call SETIC

B. Input

1. CØMMØN

TEPØCH	Minutes from midnight to epoch
DCFLG	Flags corresponding to columns 41-50 of the JDC card
TSTEP	Starting step size for the numerical integration in minutes
TICRT	x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ of the vehicle at epoch in earth radii and earth radii per minute
NDPR	Total number of Category 1 variables

2. Calling sequence

—

C. Output

1. CØMMØN

TMINUS	Flag indicating backward integration
PMAT VMAT	Arrays used in variational equation formulation, initialized at 0
TG	Time to integrate to

TUBSEF	End of file flag for observation tape
PLSTSN	"First time through" flag for RADR subroutine
FLVE	Flag for variational equations computation
TCRASH	Impact flag
IPFRST	Flag to indicate presence of an *a priori* $A^T A$
TLIST	Numerical integration working storage
DFL	Flag for DYNAT subroutine indicating first entrance

2. Calling sequence

   —

D. Error/action messages

   —

SUBROUTINES USED

A. Library

   —

B. Program

   DAUX
   VPERT

EQUATIONS

None

SUBROUTINE IDENTIFICATION

A. Title

SETIC

B. Segment

ESPØDEPH

C. Called by subroutine

PØSTPR (ESPØDEPH)
SELECT (ESPØDEPH)

FUNCTION

The function is to initialize the integration list and other parameters which must be re-initialized each time the integration is re-started.

USAGE

A. Calling sequence

Call SETIC

B. Input

1. CØMMØN

TEPØCH	Minutes from midnight to epoch
TSTEP	Starting step size for the numerical integration in minutes
TICRT	x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$ of the vehicle at epoch in Earth radii and Earth radii per minute

2. Calling sequence

—

C. Output

1. CØMMØN

TMINUS	Flag indicating backward integration
PMAT VMAT	Arrays used in variational equation formulation, initialized at 0
TG	Time to integrate to
FLVE	Flag for variational equations computation
TCRASH	Impact flag

TLIST	Numerical integration working storage
DFL	Flag for DYNAT subroutine indicating first entrance

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

DAUX
VPERT

EQUATIONS

None

SUBROUTINE IDENTIFICATION

A. Title

SETTAB

B. Segment

ESPØD

C. Called by subroutine

Driver

FUNCTION

Function is to set up the IVSTR(NIDP), IVSTR(NPRCD), VSTR(NPBIS), VSTR(NSCALE), VSTR(NBDNS), and DTMP tables.

USAGE

A. Calling sequence

Call SETTAB

B. Input

1. CØMMØN

CBØUND	Nominal set of bounds
CDEG	Degree/radian
CKMER	km/e.r.
CLDSTR	Cold-start, non-cold-start flag
CMTER	Meter/e. r.
DATA	Input storage
FGBNDS	Flag to indicate BNDS cards read
FGCAT1	Flag to indicate Category 1 card read
FGCAT2	Flag to indicate Category 2 card read
NBDNS	Starting location for the bounds used by LEGS
NDPR	Number of all differential + initial parameters to solve for (Category 1)
NIDP	Identifier for table indicating Category 1 type variables to be solved for

NPBIS — Identifies table for current estimates of Category 2 variables

NPR — Number of all parameters to solve for

NPRCD — Identifies table for definition of Category 2 variables to be solved for

NSCALE — Identifies the starting location of the list of conversion factors which convert all solution vectors and associated matrices from machine units to output units

2. Calling sequence

—

C. Output

1. CØMMØN

DTMP — Saves station number and code word for those stations with code word $\neq 0$

*IVSTR — Fixed point variable storage

*VSTR — Floating point variable storage

C. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

---

*See section on variable storage usage.

SUBROUTINE IDENTIFICATION

A. Title

SKIPT

B. Segment

ESPØDDC

C. Called by subroutines

UBSGET

FUNCTION

This subroutine skips CØMMØN portion of the observation tape after each iteration.

USAGE

A. Calling sequence

Call SKIPT

B. Input

1. CØMMØN

DBUFS	Auxiliary buffer storage
MT	Observation tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SNØMIC

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

The function is to move input initial conditions from DATA(1-12) to either TNØMX, TNØMP or TMNEL depending on whether or not DTYPE = 1., 2., 3., or 4.

USAGE

A. Calling sequence

Call SNØMIC

B. Input

1. CØMMØN

DATA	Input storage
DTYPE	Initial conditions type

2. Calling sequence

—

C. Output

1. CØMMØN

TMNEL	Initial seven-card element set
TNØMP	Initial spherical coordinates
TNØMX	Initial Cartesian coordinates

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

SNSGET

B. Segment

ESPØD

C. Called by subroutine

LØDSEN

FUNCTION

This subroutine loads sensor information from S file of SEAI tape.

USAGE

A. Calling sequence

Call SNSGET (ID, FLAG)

B. Input

1. CØMMØN

CDEG	Degrees/radian
CMTER	Meters/e.r.
DBUFS	Auxiliary buffer storage

2. Calling sequence

ID	Sensor number to be found
FLAG	Sensor flag FLAG = -1., sensor not found FLAG = 1., sensor found

C. Output

1. CØMMØN

TEMP(1)	Sensor number
(2)	ϕ (latitude, degrees)
(3)	λ (longitude, degrees)
(4)	h (altitude, meters)
(6), (7)	Sensor number
(9)	Classified or unclassified flag

2. Calling sequence

—

D. Error/action messages

1. On-line comment

   "TAPE 04 BAD - MOUNT BACKUP."
   "TYPE - GO RETRY TAPE, STØP NEXT CASE."

2. Action

   Subroutine error

SUBROUTINES USED

A. Library

   STØPGØ
   SYS
   SYSIØ
   SYSNØ
   TAPCK
   TAPEØUT

B. Program

ERRØR	Error subroutine
FLEX	Flexowriter print routine
IDSUB	Strip blanks from ID

SUBROUTINE IDENTIFICATION

A. Title
SSTB

B. Segment
ESPØDDC

C. Called by subroutines
RADR

FUNCTION

This subroutine accumulates sum, sum of squares, and number of residuals by sensor and data type. The only residuals involved are those which have satisfied both tests made in REJECT.

USAGE

A. Calling sequence
Call SSTB

B. Input

1. CØMMØN

IRCNT	Cells for partials print
NSSTB	Identifies the starting location where station information concerning computed sigmas and means of residuals are stored
PDELFG	Cells for partials print
PRESD	Residuals (ΔR, ΔA, ΔE, $\Delta\dot{R}$, ΔHA, ΔDEC)
PUBS	Sensor number, time, R, A, E, $\dot{R}$, α, δ table
TEMP	Temporary storage
VSTR	Floating point variable storage
CDEG	Degrees/radians
CKMER	km/e.r.

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling Sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

METHOD

VSTR (NSSTB+0) = sensor number

VSTR (NSSTB+1) $= \Sigma(\Delta R_i)$

VSTR (NSSTB+2) $= \Sigma(\Delta R_i)^2$

VSTR (NSSTB+3) $= N_R^{a} * 10000. + N_R^{R}$ where $\begin{cases} \text{a super = accepted} \\ \text{R super = rejected} \end{cases}$

VSTR (NSSTB+4) $= \Sigma(\Delta A_i)$

VSTR (NSSTB+5) $= \Sigma(\Delta A_i)^2$

VSTR (NSSTB+6) $= N_A^{a} * 10000. + N_A^{R}$

VSTR (NSSTB+7) $= \Sigma(\Delta E_i)$

VSTR (NSSTB+8) $= \Sigma(\Delta E_i)^2$

VSTR (NSSTB+9) $= N_E^{a} * 10000. + N_E^{R}$

VSTR (NSSTB+10) $= \Sigma(\Delta \dot{R}_i)$

VSTR (NSSTB+11) $= \Sigma(\Delta \dot{R}_i)^2$

VSTR (NSSTB+12) $= N_{\dot{R}}^{a} * 10000. + N_{\dot{R}}^{R}$

VSTR (NSSTB+13) = (next) sensor number

.
.
.

etc.

SUBROUTINE IDENTIFICATION

A. Title

STSMAT

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

The function is to convert the upper triangular S matrix in DATA (321 - 520) from human units to machine units and then transfer to VSTR (NATA) in the special way described under equations.

USAGE

A. Calling sequence

Call STSMAT

B. Input

1. CØMMØN

DATA (321-520)	Input data
DCFLG	ESPØDDC control flags
NPR	Number of all parameters to solve for
NATA	Identifies the starting location of where the triangular A^TA is stored
NSCALE	Identifies the starting location of the list of conversion factors which convert all solution vectors and associated matrices from machine units to output units

2. Calling sequence

—

C. Output

1. CØMMØN

VSTR (NATA)	Identifies the starting location of where the upper triangular S matrix is stored in variable storage

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

HUMAH — Converts vector or matrix from machine units to human units or vice versa

EQUATIONS

```
         I = 1
        II = 1
    DØ 2 J = 1, NPR
        JJ = NPR - J + 1
    DØ 1 K = 1, JJ
        KK = NATA + II -1
  VSTR(KK) = DATA (I + 320)
        II = II + 1
         I = I + 1
        II = II + 1
  DCFLG(2) = 1
```

SUBROUTINE IDENTIFICATION

A. Title

SUPMAT

B. Segment

ESPØD

C. Called by subroutine

DRIVER

FUNCTION

The function is to move the initial update matrix from temporary storage to permanent storage VSTR (NRTMP) and convert from human units to machine units.

USAGE

A. Calling sequence

Call SUPMAT

B. Input

1. CØMMØN

DATA	Input storage
NPR	Number of all parameters to solve for
NRTMP	Identifies the starting location of temporary storage for special handling of the R matrix
NSCALE	Identifies the starting location of the list of conversion factors which convert all solution vectors and associated matrices from machine units to output units

2. Calling sequence

—

C. Output

1. CØMMØN

VSTR (NRTMP)	Identifies the starting location of storage for the initial update matrix

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

HUMAH	Converts vector or matrix from machine units to human units or vice versa

SUBROUTINE IDENTIFICATION

A. Title

SWTSN

B. Segment

ESPØD

C. Called by subroutine

LØDØBS

FUNCTION

For a given observation set this subroutine monitors the following:

a) Observation weight assignment

b) Refraction corrections to elevation angles if called for

c) Precession of optical data if necessary

d) Formatting of the observation set into the format to be written on tape

e) Recording of the sensor number of this observation set to insure that information regarding the position of this sensor will be included in the master sensor table when this table is established.

USAGE

A. Calling sequence

Call SWTSN (A, I)

B. Input

1. CØMMØN

CSIG	Sensor sigmas
CSTYPE	Sensor type for σ, $\overline{N}_s$ and N
DBUFS	Auxiliary buffer storage
TEMP	Temporary storage

2. Calling sequence

A(I)	Starting location of a ten word array containing the observation set

C. Output

See tape format

D. Error/action messages

1. Off-line comment

"REQUIRED SENSORS TABLE FULL. ITEM NOT SAVED IS ___"

2. Action

None

SUBROUTINES USED

A. Library

GLØP
XSRCH

B. Program

CALCSG	Compute σ_R, σ_A, σ_E, σ_{RDT} from credance
PRECES	Computes precession of optical data
REFRAC	Computes tropospheric refraction correction

WORD NO. 1	X	X	T	T	O	N	N	N
WORD NO. 2	M	M	M	M	M	S	S	S
WORD NO. 3	48 BIT FLOATING POINT NO.							
WORD NO. 4	R	Ø	Ø	Ø	Ø	Ø	E	A
WORD NO. 5	48 BIT FLOATING POINT NO.							
WORD NO. 6	48 BIT FLOATING POINT NO.							
WORD NO. 7	48 BIT FLOATING POINT NO.							
WORD NO. 8	48 BIT FLOATING POINT NO.							
WORD NO. 9	X X X			MAX		MIN		INT
WORD NO. 10	CF	CL	ØT	ØBS. NØ				

WORD NO. 1

FILE MAINTENANCE CONTROL

EQUIPMENT TYPE

STATION NUMBER

X	X	T	T	O	N	N	N

WORD NO. 2 MESSAGE AND SATELLITE NUMBERS

WORD NO. 3 TIME IN DAYS SINCE 1950

WORD NO. 4 R-ASSOCIATION INDICATOR A - ACCURACY

E - EQUINOX Ø - NOT USED

WORD NO. 5 ELEV - DEC

WORD NO. 6 AZ - RA

WORD NO. 7 SLANT RANGE

WORD NO. 8 RANGE RATE OR MAX FREQUENCY SHIFT

WORD NO. 9 BRIGHTNESS AT TIME OF OBSERVATION (BCD)

X X X	MAX	MIN	INT

10 BIT BINARY INTEGERS

THIS WORD IS ALL BLANKS IF BRIGHTNESS IS NOT REPORTED

Figure 4-19 a. Tape Format — 10-Word Array Entering SWTSN

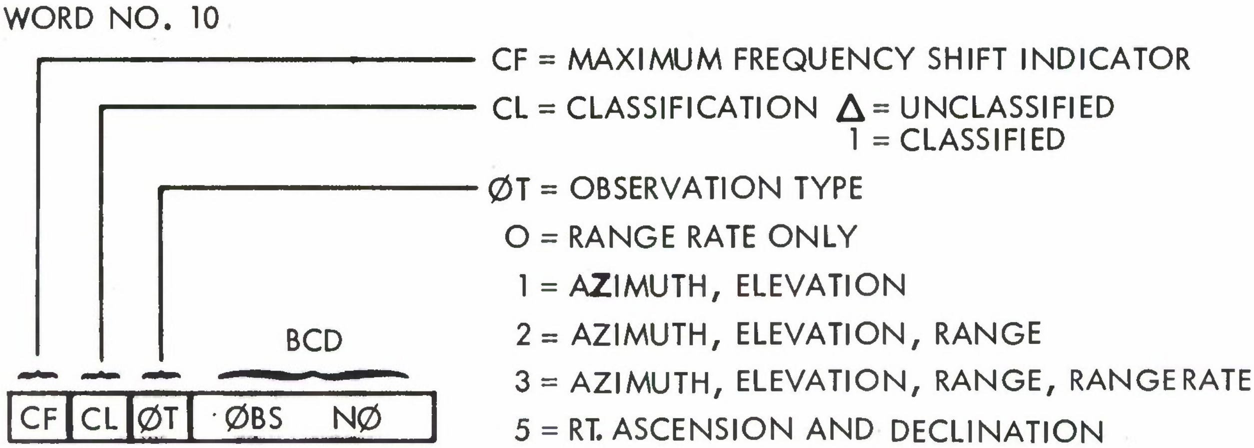


Figure 4-19 b. Tape Format — 10-Word Array Entering SWTSN (Continued)

WORD NO. 1	G_s - BINARY INTEGER AT B23			O	N	N	N		G_s; SENSOR NUMBER (BCD)
WORD NO. 2	CF	CL	ϕT	E	T	S	S	S	(SEE INDEX 1 BELOW)
WORD NO. 3	48 BIT FLOATING POINT NO.								TIME - DAYS AND FRACTIONS OF DAYS FROM JAN 1, 1950
WORD NO. 4	R	Ø	Ø	Ø	Ø	Ø	E	A	(SEE INDEX 2 BELOW)
WORD NO. 5	48 BIT FLOATING POINT NO.								ELEVATION - DECLINATION
WORD NO. 6	48 BIT FLOATING POINT NO.								AZIMUTH - HOUR ANGLE
WORD NO. 7	48 BIT FLOATING POINT NO.								SLANT RANGE
WORD NO. 8	48 BIT FLOATING POINT NO.								RANGE RATE
WORD NO. 9	σ_R				σ_A				*OBSERVATION WEIGHTS ASSIGNED AT OBSERVATION PROCESSING TIME
WORD NO. 10	σ_E				$\sigma_{\dot{R}}$				

INDEX 1

CF = MAXIMUM FREQUENCY SHIFT INDICATOR

CL = CLASSIFICATION Δ = UNCLASSIFIED
I = CLASSIFIED

ϕT = OBSERVATION TYPE

- 0 RANGE RATE ONLY
- 1 AZIMUTH AND ELEVATION
- 2 AZIMUTH ELEVATION AND RANGE
- 3 AZIMUTH, ELEVATION, RANGE AND RANGE RATE
- 5 RIGHT ASCENSION AND DECLINATI ON

ET = EQUIPMENT TYPE

A = ACCURACY

INDEX 2

R = ASSOCIATION INDICATOR

E = EQUINOX

A = ACCURACY

*THESE WEIGHTS ARE STORED AS BINARY INTEGERS, TWO PER WORD (ONE AT A B23 AND THE OTHER AT A B47). THE TRUE WEIGHTS ARE THESE INTEGERS CONVERTED TO FLOATING POINT NUMBERS AND THE DIVIDED BY 10^4. FOR OPTICAL DATA THE FIRST WORD CONTAINS WEIGHTS FOR FIELD REDUCED RA AND DEC AND THE SECOND WORD CONTAINS WEIGHTS FOR PRECISION REDUCED RA AND DEC.

Figure 4-20. 10-Word Array Leaving SWSN in A(I)

SUBROUTINE IDENTIFICATION

A. Title

TCØMP

B. Segment

ESPØDEPH

C. Called by subroutine

PØSTPR

FUNCTION

This subroutine compares $|(x) - x_T, (y) - y_T, (z) - z_T|$ with ϵ. Four sets of x_T, y_T, z_T, t_D, t_{FD} are the maximum that can be handled. The results of the comparison are printed with the ESPØDEPH output under the heading "CØMPARISØN PØINT (K)."

USAGE

A. Calling sequence

Call TCØMP (K)

B. Input

1. CØMMØN

DTARG	Array containing the four sets of x_T, y_T, z_T, t_D, t_{FD}
TEMP	Temporary storage
TRAJX	Array containing the integrated x, y, z at times $(t_D + t_{FD})$
CKMER	km/e.r. conversion factor
KØUT	Output tape number

2. Calling sequence

K	Set number (i.e., K = 1, 2, 3, 4)

C. Output

1. CØMMØN

Sets of x, y, z, t_D, t_{FD} are set at octal location 15100 to be used as inputs by the GIPAR program.

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

1. Library

GLØP
SQRTF

B. Program

ØUTPT

SUBROUTINE IDENTIFICATION

A. Title

TGDJD

B. Segment

ESPØDEPH

C. Called by subroutine

TPRNT

FUNCTION

Function is to compute Julian date and calendar date from integration time and prints date line of trajectory block print.

USAGE

A. Calling sequence

Call TGDJD

B. Input

1. CØMMØN

TEMP	Temporary storage
TEPØCH	Epoch time, minutes from midnight
TG	Time to integrate to (from 0^h day of epoch)

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP
PANT

B. Program

DATE - sets up calendar date from TG

SUBROUTINE IDENTIFICATION

A. Title

TIME

B. Segment

ESPØD

C. Called by subroutines

ØBSIN
TINIT

FUNCTION

Function is to compute the Julian date. Year and TMØNTH must be whole numbers. Day may be a fraction (fractional days will be added into TØMIN, THØUR; TMINS and SECS may be fractional).

USAGE

A. Calling sequence

Call TIME (YEAR, TMØNTH, DAY, THØUR, TMINS, SECS, TJD, TØTMN)

B. Input

1. CØMMØN

—

2. Calling sequence

YEAR	Two digits (i.e., 63) (year)
TMNTH	(Month)
DAY	Any number of digits (day)
THØUR	Any number of digits (hour)
TMINS	Any number of digits (minutes)
SECS	Any number of digits (seconds)

C. Output

1. CØMMØN

—

2. Calling sequence

TJD	Julian date
TØTMN	Number of minutes after midnight

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

EQUATIONS

Julian Date

$$JD = 2433282.5 + [365\,(YY - 50) + [n] + (M + DD - 1)]$$

To obtain the number of days additional due to leap year

$$[n] = \frac{YY - 48}{4}$$

where $[n]$ = integer part only; if there is no remainder then

$$[n] = [n] - 1$$

since YY is now the beginning of a leap year.

YY = Last two digits of epoch year

M = Number of days in previous months from 1 January

DD = Day of epoch month

This equation calculates the Julian date for 1 January of the specified year; the following examples show the remainder of the calculation:

Example 1: 21 March 1965

Y - 1950 = 15 (365) (15) = 5475

Y - 1948 = 17 $[n]$ = 4, not a leap year

JD = 2438761.5 + 59 + 20 = 2438840.5

59 from table of days in previous months since 1 January (not including month of interest)

20 = (DD - 1)

Example 2: 20 July 1980

$$Y - 1950 = 30 \qquad (365)(30) = 10950$$

$$Y - 1948 = 32 \qquad [n] = [7], \text{ a leap year}$$

$$JD = 244239.5 + 182 = 19 = 2444440.5$$

SUBROUTINE IDENTIFICATION

A. Title

TINIT

B. Segment

ESPØD

C. Called by subroutine

DPRIM
MNELTC

FUNCTION

Function is to take the epoch time and find the Julian date; the epoch time is in minutes from midnight of epoch day and to compute α_{go}.

USAGE

A. Calling sequence

Call TINIT

B. Input

1. CØMMØN

CDAYMN	Number of days in month
CDEG	Degrees/radian
CPI	π
C2PI	2π
DDAY	Epoch day
DHØUR	Epoch hour
DMIN	Epoch minute
DSEC	Epoch second
DTYPE	Initial condition type
DYEAR	Epoch year
TEMP	Temporary storage

2. Calling sequence

—

C. Output

1. CØMMØN

CSEPS	cos ϵ
DLEPS	$\Delta\epsilon$
DLPSI	$\Delta\psi$
DNUT	Nutation correction
DSDAY	Epoch day, days from beginning of year
DSFDAY	Epoch time, fraction of day

TALFAG	α_g for midnight day of epoch
TEPØCH	Epoch time, minute from midnight
TJDATE	Julian date of midnight, epoch day

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF
SINF

B. Program

PIMOD	Takes principal value of angle between 0 and 2π
TIME	Computes Julian date

EQUATIONS

1. N, Nutation Correction

$\epsilon = 23^{\circ}\ 26'\ 35''$
$\cos\ \epsilon = 0.9174559$
J = number of years from 1900
$\Omega = 259^{\circ}\!.18 - 19^{\circ}\!.3414\ (J)$
$\Delta\psi = -17''\!.2 \sin\Omega + 1''\!.3 \sin\left[-160^{\circ}\!.61 + 719^{\circ}\!.9957\ (J)\right]$
$N = \cos\ \epsilon\, \Delta\psi$

2. $\Delta\epsilon = 9''\!.2 \cos\Omega + 0''\!.6 \cos\left[-160^{\circ}\!.61 + 719^{\circ}\!.9957\ (J)\right]$

3. α_{go} at epoch (true of date equinox and equator)

J_o = Julian date -2,430,000

$$J = \frac{J_o + 14,980}{365.25}$$

$[J]$ = Integer part of J only

$$\alpha_{go} = \left\{\frac{\pi}{43200}\left[23925.836 + 1.84542J + \left(9.29 \times 10^{-6}\right)J^2 + N\right] + 2\pi\left(J - [J]\right)\right\} \quad \text{(radian)}$$

$$0 \le \alpha_{go} \le 2\pi$$

4. Days (fractional portion) since 01 Jan, year of epoch

$$\text{Fractional Days} = \frac{\text{D hour}}{24} + \frac{\text{D min}}{1440} + \frac{\text{D sec}}{86,400}$$

SUBROUTINE IDENTIFICATION

A. Title

TMSEP

B. Segment

ESPØD

C. Called by subroutine

DPRLM
MNELTC

FUNCTION

Function is to convert the year, number of days from beginning of year, and the fraction of a day to the year, month, day, hour, minute, and second. This routine does not account for the possibility of a non-leap year at the turn of the century (e. g. , 2000 A. D.).

USAGE

A. Calling sequence

Call TMSEP

B. Input

1. CØMMØN

CDAYMN	Number of days in month
DSDAY	Epoch day, days from beginning of year
DSFDAY	Epoch time, fraction of day
DYEAR	Year from beginning of the century (e. g. , 50, 59, 63)

2. Calling sequence

—

C. Output

1. CØMMØN

DDAY	Epoch day
DHØUR	Epoch hour
DMIN	Epoch minute
DMNTH	Epoch month
DSEC	Epoch second

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

TPRLM

B. Segment

ESPØDDC

C. Called by subroutine

INTEG

FUNCTION

Function is to perform the initialization necessary prior to the start of the first iteration of a differential correction run. The auxiliary quantity α used in the simulation of radiation pressure is computed here.

USAGE

A. Calling sequence

Call TPRLM

B. Input

1. CØMMØN

TNØMX	$x, y, z, \dot{x}, \dot{y}, \dot{z}$ of the vehicle at epoch (km, km/sec)
TNØMP	α, δ, β, A, R, v of the vehicle at epoch (km, km/sec, deg)
CDEG	Degrees per radian
CDTER	Conversion from input distance unit to Earth radii
CVTERM	Conversion from input velocity unit to Earth radii per minute
CSØLC	Solar constant (watts/meter^2)
CLIGHT	Speed of light (km/sec)
CERAU	Earth radii per astronomical unit
DAREA	Effective area of the spacecraft (meter^2)
DMASS	Mass of the spacecraft (kg)
CKMER	Kilometers per Earth radii
NDPR	Number of Category 1 variable
NIDP	Locator in IVSTR of the table of identifiers of the Category 1 variables
CDAD2M	Value of $C_D A/2m$
CK	Value of K
NPR	Total number of Category 1 and 2 variables being solved for
NPBIS	Locator in VSTR of the initial bias estimates of the Category 2 variables

2. Calling sequence

—

C. Output

1. CØMMØN

TICRT	$x, y, z, \dot{x}, \dot{y}, \dot{z}$ of the vehicle at epoch (e. r., e. r. / min)
TIPØL	$\alpha, \delta, \beta, A, R, v$ of the vehicle at epoch (e. r., e. r. / min radians)
TALFA	α, used in RPRESS subroutine
CENTER	Central body in integration (0 for Earth)
RJUPT	Test radius for inclusion of Jupiter as a perturbation influence on trajectory (set to 156. ER)
TSUSB	Best RMS in curve fit (initialized at 10^{33})
NITCT	Iteration counter (set to 1)
VSTR(NPAR)	Initial values of the solution parameters

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

JCS

EQUATIONS

$$\alpha = \frac{S}{C^2} \cdot R^2 \frac{A}{m}$$

where

S = CSØLC
C = CLIGHT
R = CERAU
A = DAREA
m = DMASS

SUBROUTINE IDENTIFICATION

A. Title

TPRLM

B. Segment

ESPØDEPH

C. Called by subroutines

PØSTPR (ESPØDEPH)

FUNCTION

Function is to perform the initialization necessary prior to the start of the trajectory simulation. The auxiliary quantity α used in the simulation of radiation pressure is computed here.

USAGE

A. Calling sequence

Call TPRLM

B. Input

1. CØMMØN

TNØMX	$x, y, z, \dot{x}, \dot{y}, \dot{z}$ of the vehicle at epoch (km/sec, km)
TNØMP	α, δ, β, A, R, v of the vehicle at epoch (km/sec, km, deg)
CDEG	Degrees per radian
CDTER	Converts input distance unit to Earth radii
CVTERM	Converts input velocity unit to Earth radii per minute
CSØLC	Solar constant (watts/m^2)
CLIGHT	Speed of light (km/sec)
CERAU	Earth radii per astronomical unit
DAREA	Effective area of the spacecraft (meters^2)
DMASS	Mass of the spacecraft (kg)
CKMER	Kilometers per Earth radii

2. Calling sequence

—

C. Output

1. CØMMØN

TICRT	$x, y, z, \dot{x}, \dot{y}, \dot{z}$ of the vehicle at epoch (Earth radii, e.r./min)
TIPØL	$x, y, z, \dot{x}, \dot{y}, \dot{z}$ of the vehicle at epoch (Earth radii, e.r./min, radians)

TALFA	α, used in RPRESS subroutine
CENTER	Central body in integration (0 for Earth)
RJUPT	Test radians for inclusion of Jupiter as a perturbative influence on trajectory (set to 156. e. r.)
NDTCT	Counter used in SELECT for Δt, T table

D. Error/action messages

—

SUBROUTINES USED

A. Library

—

B. Program

JCS

EQUATIONS

$$\alpha = \frac{S}{C^2} \cdot R^2 \cdot \frac{A}{m}$$

where

S = CSØLC
C = CLIGHT
R = CERAU
A = DAREA
M = DMASS

SUBROUTINE IDENTIFICATION

A. Title

TPRNT

B. Segment

ESPØDEPH

C. Called by subroutine

PØSTPR

FUNCTION

Function is to set up and execute the printing of trajectory information in ESPØDEPH.

USAGE

A. Calling sequence

Call TPRNT

B. Input

1. CØMMØN

TALFAG	α_g for midnight day of epoch
TEMP	Temporary storage
TEPØCH	Epoch time, minutes from midnight
TG	Time to integrate to
TRAJX	$x, y, z, \dot{x}, \dot{y}, \dot{z}$ at time TG
CDEG	Degrees/radian
CELLIP	Ellipticity of the Earth
CKMER	km/e.r.
CMU	GM Earth (e.r.^{3}/min^2)
CPI	π
C2PI	2π
CWE	Earth's rotational rate
IØUT	Output tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP
PANT
SQRTF

B. Program

ATNQF	Arc tangent
CTØP	Converts cartesian to polar
PIMØD	Principal angle between 0 and 2π
PLTEL	Converts polar to elements
TGDJD	Computes Julian date to calender date from integration time

EQUATIONS

Compute latitude, longitude and altitude.

Geodetic latitude, $\Phi^* = \tan^{-1}\left[\dfrac{z}{\left(x^2 + y^2\right)^{1/2} (1 - \epsilon)^2}\right]$

Longitude, $\lambda = \alpha - \alpha_{go} - \omega_e t$

Height, $h = r - \dfrac{a_e (1 - \epsilon)}{\left[1 - (2\epsilon - \epsilon^2) \dfrac{x^2 + y^2}{r^2}\right]^{1/2}}$

Compute apogee and perigee altitudes.

Apogee altitude (e. r.), $AP_h = a^* (1 + e) - a_e$

Perigee altitude (e. r.), $P_h = a^* (1 - 3) - a_e$

Compute the period and time of the next nodal crossing measured in minutes from epoch.

$$P = \frac{2\pi a^{3/2}}{\mu^{1/2}} \qquad n = \frac{\mu^{1/2}}{a^{3/2}}$$

$$\cos E_N = \frac{r \cos(-\omega) + a_e}{a}$$

ω = argument of perigee

E_N = eccentric anomally of the nodal crossing

$$\sin E_N = \sqrt{1 - \cos^2 E_N}\,, \qquad E_N = \tan^{-1}\left[\frac{\sin E_N}{\cos E_N}\right]$$

$$M_N = E_N - e \sin E_N$$

$$\Delta t = \frac{M_N - M_o}{n}$$

If $\Delta t < 0$, set $\Delta t = \Delta t + P$

$$T_N = T + \Delta t - T_o$$

where

T = Current time (min from 0^h day of epoch)

T_o = Epoch time (min from 0^h day of epoch)

SUBROUTINE IDENTIFICATION

A. Title

TRAJ

B. Segment

ESPØDDC
ESPØDEPH

C. Called by subroutine

PØSTPR (ESPØDEPH)
INTEG (ESPØDDC)

FUNCTION

This subroutine integrates the equations of motion and variational equations to a specified time.

USAGE

A. Calling sequence

Call TRAJ(TN)

B. Input

1. CØMMØN

   (See storage allocation on next page)

2. Calling sequence

   TN Time to integrate to

C. Output

1. CØMMØN

   (See storage allocation on next page)

2. Calling sequence

   —

SUBROUTINES USED

A. Program

DAUX

Table 4-IV. CØMMON (YYYY) Storage

YYYY	Program Tag	Description
(851)	NDPR	Number of variational parameters
(479)	HMAX	Maximum allowable step size
(480)	HMIN	Minimum allowable step size
(482)	ER	Step size test parameters
(481)	YMIN	See method
(483)	NRRR	Ratio of Runge-Kutta step to Cowell step
(861)	FLVE	Flag for DAUX routine
(862)	SKIP	= 0, skip second evaluation of $\ddot{Y}$ for var. parameters
(1438)	TRAJX	Output— contains values consistent with TN
(1495)	TLIST	Input and storage, at output values consistent with T

T may equal TN, but their difference will never be greater than H.

TRAJX(1-3)	x, y, z
TRAJX(4-6)	$\dot{x}$, $\dot{y}$, $\dot{z}$
TRAJX(7-9)	$\ddot{x}$, $\ddot{y}$, $\ddot{z}$
TRAJX(10-15)	$\delta_1 x$, $\delta_1 y$, $\delta_1 z$, $\delta_1 \dot{x}$, $\delta_1 \dot{y}$, $\delta_1 \dot{z}$ first variation
TRAJX(16-21)	$\delta_2 x$, $\delta_2 y$, $\delta_2 z$, $\delta_2 \dot{x}$, $\delta_2 \dot{y}$, $\delta_2 \dot{z}$ second variation
	etc. for NDPR variations

Table 4-V. CØMMØN (TLIST) Storage

TLIST	Program Tag	Description			
1	FLAG	Initialization parameter—initialize when nonzero			
2	T	Current time	These values must be supplied when FLAG ≠ 0		
3	H	Current step size			
4-30	Y(1-27)	$y_1, y_2, \cdots, y_n$			
31-57	YP(1-27)	$\dot{y}_1, \dot{y}_2, \cdots, \dot{y}_n$			
58-84	YPP(1-27)	$\ddot{y}_1, \ddot{y}_2, \cdots, \ddot{y}_n$ DAUX stores 2nd der.			
85-192	TR(1-27, 1-4)	Intermediate storage	N = 3(NDPR + 1)		
193-489	DIF	Difference table		During Runge-Kutta phase	
	(1, 1-27)	$\nabla^8 f_i$	as I = 1, N	$\ddot{y}_{io}$	as I = 1, N
	(2, 1-27)	$\nabla^7 f_i$		$\ddot{y}_{i1}$	
	(3, 1-27)	$\nabla^6 f_i$		$\ddot{y}_{i2}$	
	(4, 1-27)	$\nabla^5 f_i$		$\ddot{y}_{i3}$	
	(5, 1-27)	$\nabla^4 f_i$	See method for description of this table.	$\ddot{y}_{i4}$	These values are saved during 8NR Runge Kutta steps.
	(6, 1-27)	$\nabla^3 f_i$		$\ddot{y}_{i5}$	
	(7, 1-27)	$\nabla^2 f_i$		$\ddot{y}_{i6}$	
	(8, 1-27)	$\nabla^1 f_i$		$\ddot{y}_{i7}$	
	(9, 1-27)	$f_i = y$		$\ddot{y}_{i8}$	
	(10, 1-27)	$'F_i$		y_{i4}	
	(11, 1-27)	$''F_i$		$\dot{y}_{i4}$	

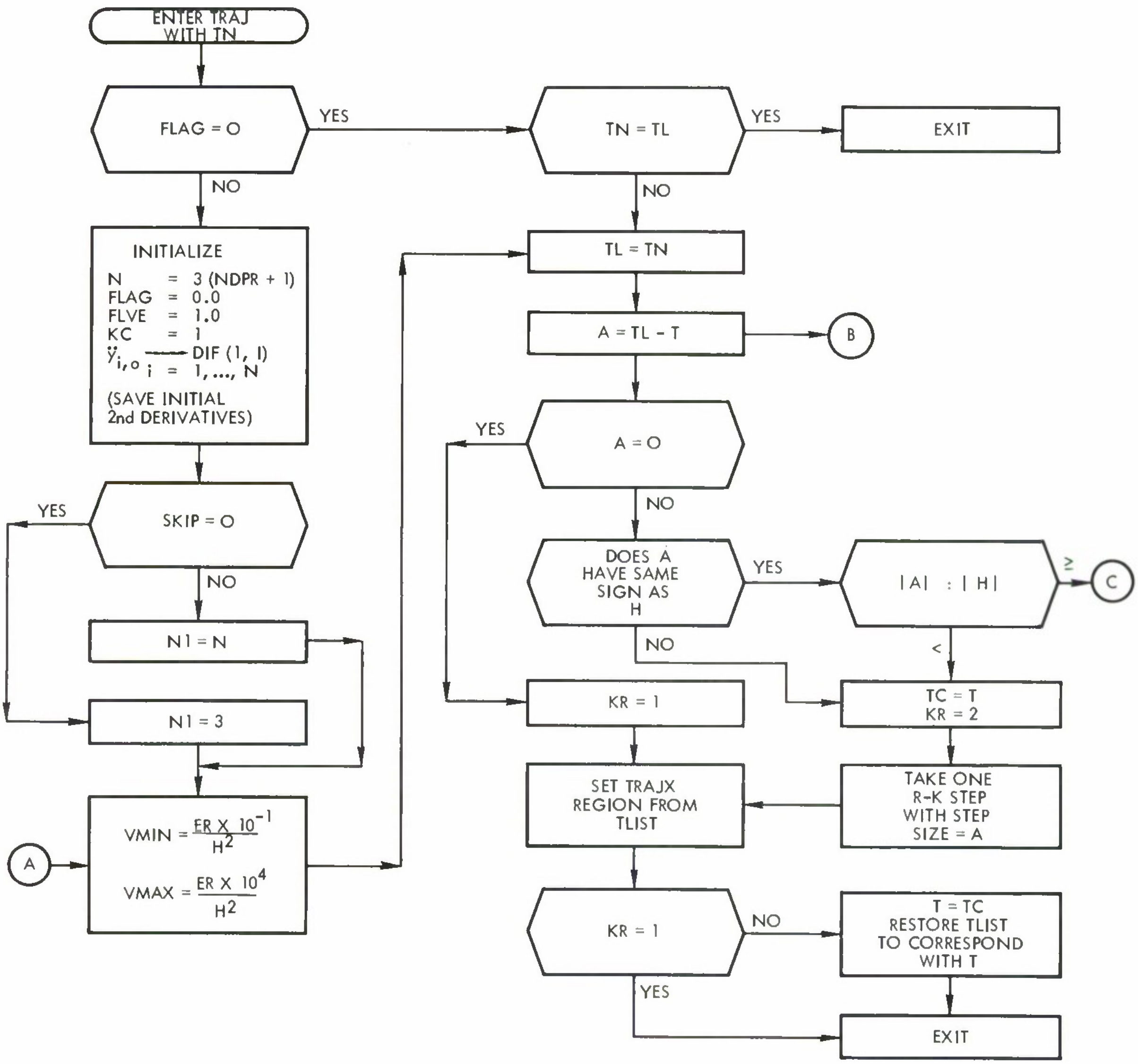


Figure 4-21 a. TRAJ Flow Diagram

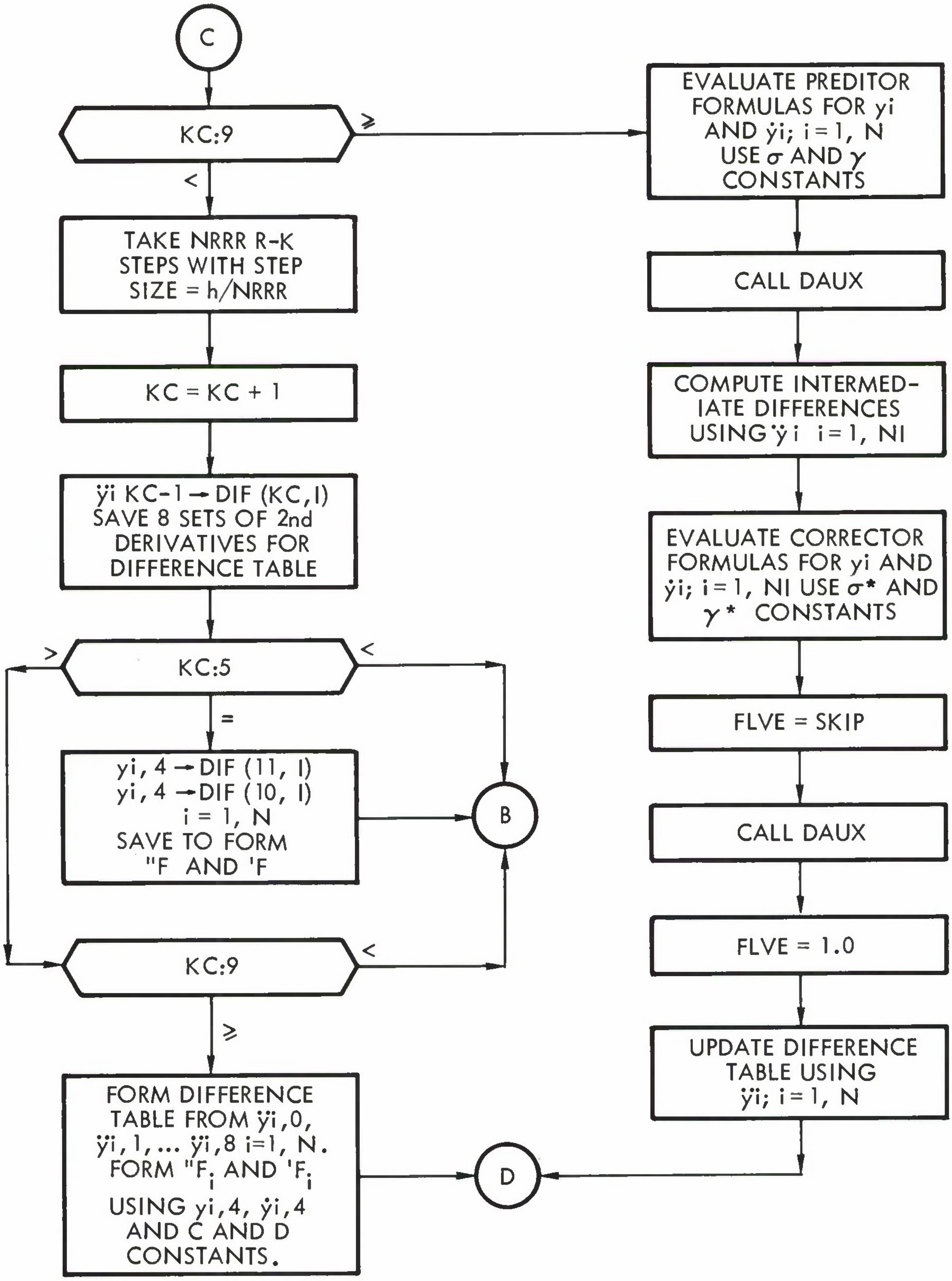


Figure 4-21 b. TRAJ Flow Diagram (Continued)

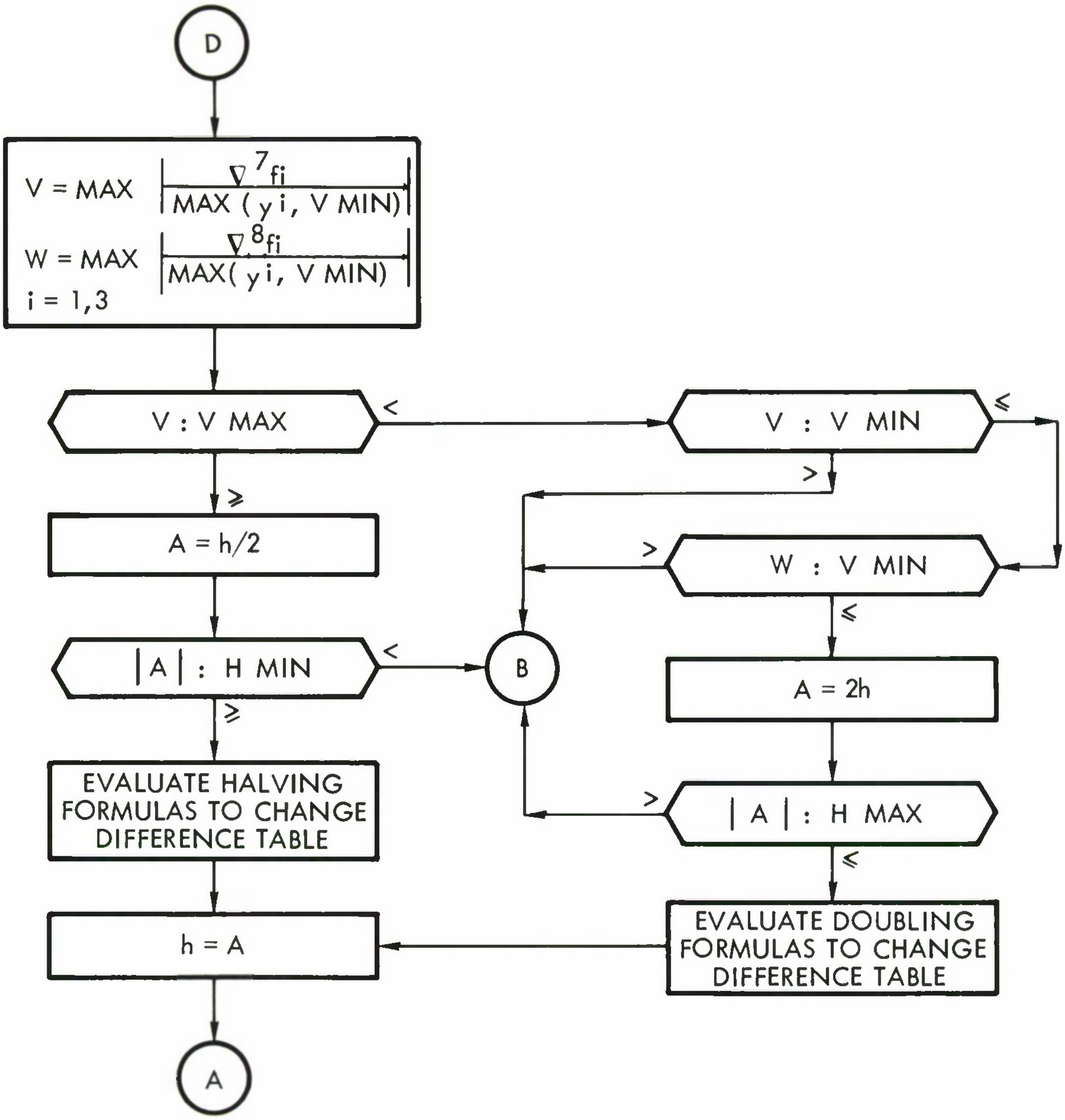


Figure 4-21 c. TRAJ Flow Diagram (Continued)

SUBROUTINE IDENTIFICATION

A. Title

TTAPE

B. Segment

ESPØDEPH

C. Called by subroutine

PØSTPR

FUNCTION

Function is to generate a trajectory tape of x, y, z, $\dot{x}$, $\dot{y}$, and $\dot{z}$ vs T. Where T is minutes from 0^h day of epoch, x, y, z are in Earth radii, and $\dot{x}$, $\dot{y}$, $\dot{z}$ are in Earth radii per KEMIN.

USAGE

A. Calling sequence

Call TTAPE

B. Input

1. CØMMØN

CMU	GM Earth (e.r.^{3}/min^2)
DBUFS	Auxiliary buffer storage
DVEHN	Vehicle name and number (BCD)
IØUT	Output tape number
TG	Time to integrate to
TRAJX(1)	x (e. r.)
(2)	y (e. r.)
(3)	z (e. r.)
(4)	$\dot{x}$ (e. r./min)
(5)	$\dot{y}$ (e. r./min)
(6)	$\dot{z}$ (e. r./min)

2. Calling sequence

—

C. Output

The trajectory tape which is generated is made up of blocks of data, each block is made up of 18 sets, containing 7 words of the following format:

T	time in minutes from 0^h day of epoch
x	(e. r.)
y	(e. r.)
z	(e. r.)
$\dot{x}$	(e. r. /KEMIN)
$\dot{y}$	(e. r. /KEMIN)
$\dot{z}$	(e. r. /KEMIN)

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD
SQRTF

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

TWRAP

B. Segment

ESPØDEPH

C. Called by subroutine

PØSTPR

FUNCTION

Function is to write the final block of trajectory tape and the sentinel block.

USAGE

A. Calling sequence

Call TWRAP

B. Input

1. CØMMØN

DBUFS	Auxiliary buffer storage
NØUT	Ephemeris tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØrd

B. Program

TTAPE	Generates ephemeris tape ($x, y, z, \dot{x}, \dot{y}, \dot{z}$ vs t)

SUBROUTINE IDENTIFICATION

A. Title

UBRED

B. Segment

ESPØDDC

C. Called by subroutine

UBSGET

FUNCTION

This subroutine reads observations into variable storage. The observations are read from LØG 7 and into variable storage starting at location VSTR (NUBS). As many blocks will be read as can be handled in the remaining portion of CØMMØN storage.

USAGE

A. Calling sequence

Call UBRED

B. Input

1. CØMMØN

DBUFS	Auxiliary buffer storage
MT	Observation tape number

2. Calling sequence

—

C. Output

1. CØMMØN

VSTR(NUBS)	Array containing observations from LØG 7

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

UBSGET

B. Segment

ESPØDDC

C. Called by subroutine

SELECT

FUNCTION

Function is to get next observation time from variable storage.

USAGE

A. Calling sequence

Call UBSGET

B. Input

1. CØMMØN

DBUFS	Auxiliary buffer storage
VSTR (NUBS)	The starting location of the table of observations
TEMP	Tempory storage
TUBSEF	Sentinel block detection flag
CØMLST	Dimension of variable storage

2. Calling sequence

—

C. Output

1. CØMMØN

PUBS (1)	Sensor number	
(2)	Time in min from 0^h day of epoch	
(3)	Range measurement	
(4)	Azimuth measurement	
(5)	Elevation measurement	
(6)	Range rate measurement	if applicable
(7)	Hour angle measurement	if applicable
(8)	Declination measurement	if applicable

PSIG (1)	σ_R	Observation weights
(2)	σ_A	
(3)	σ_E	
(4)	σ_{RDT}	
(5)	σ_{HA}	
(6)	σ_{DECL}	
PKSUBS	Rejection criterion for this sensor's observation	

Note: These three arrays, though being considered here as outputs from this package are actually setup by routine MØVEVS which is driven by UBSGET

2. Calling sequence

—

SUBROUTINES USED

A. Library

—

B. Program

MØVEVS	Moves observation set from variable storage to working storage
REWT	Rewinds the observation tapes
SKIPT	Skips CØMMØN portion of observation tape at the beginning of all iterations except the first
UBRED	Reads observations into variable storage

SUBROUTINE IDENTIFICATION

A. Title

UNPAKSN

B. Segment

ESPØD

C. Called by subroutines

PRCØNS

FUNCTION

This subroutine unpacks 23 bit integers stored in two cells, (A) and (B) (at a B23 and B47), into four cells starting in C(1). The numbers in (C) are in floating point and scaled (i. e., C(1) = A(LEFT)/10000.).

USAGE

A. Calling sequence

Call UNPAKSN (A, B, C)

B. Input

1. CØMMØN

—

2. Calling sequence

A Two 23 bit integers
B Two 23 bit integers

C. Output

1. CØMMØN

—

2. Calling sequence

C 4 floating point numbers (scaled)

D. Error/action messages

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title
   UPDATE

B. Segment
   ESPØDEPH

C. Called by subroutine
   PØSTPR

FUNCTION

Function is to update a given covariance matrix to a specified time t and to print the resulting matrices. The covariance matrix to be updated can either be a 6 x 6 (α, δ, β, A, R, v), a 7 x 7 (α, δ, β, A, R, v, $C_D A/2m$ or K) or an 8 x 8 (α, δ, β, A, R, v, $C_D A/2m$, K).

The updated matrices are given in polar, Cartesian, and orbit plane systems.

USAGE

A. Calling sequence
   Call UPDATE

B. Input

1. CØMMØN

DCFLG	DC package control flags
DTMP	Used as temporary storage by this routine
NATA	Identifies the starting location of where the triangular $A^T A$ is stored
NBDNS	Starting location for the bounds
NDPAR1	Starting location where the solution vector will be stored
NDPR	Number of all differential and initial parameters to solve for (Category 1)
NPR	Number of all parameters to solve for
NR	Starting location of where the inverse $A^T A$ is stored

NRTMP	Starting location of temporary storage for special handling of the R matrix
NSCALE	Starting location of the list of conversion factors
PSTFLG	Post-processor control flags
PUBS	Used as temporary storage
TDPDX	Contains matrices of partials for covariance matrix update
TRAJX	Array containing x, y, z, $\dot{x}$, $\dot{y}$, $\dot{z}$, etc.
VSTR	Floating point variable storage
CKMER	km/e.r.
KØUT	Output tape number

2. Calling sequence

—

C. Output

Print—ȻPOLAR, ȻCARTESIAN, ȻORBIT PLANE, Eigenvalues and Eigenvectors corresponding to upper 3 x 3 of ȻORBIT PLANE, and rotational angles for this upper 3 x 3 into its principal axis.

D. Error/action messages

—

SUBROUTINES USED

A. Library

GLØP

PANT

B. Program

CØRMAT	Computes correlation (σ and ρ) matrix
HUMAH	Converts vector or matrix from machine units to human units or vice versa
LEGS2	Least squares package, solves Ax = B

MABAT	Multiplies ABA^T where B is a lower triangular matrix of dimension n and A is an n x n full matrix
MLTUT	Converts lower triangular matrix to upper triangular matrix
POPPC	Sets up rotation from cartesian to orbit plane coordinates
PPLPC	Computes partial of ADBARV with respect to Cartesian coordinates
PRAXIS	Computes components of the principal axis of the u, v, w covariance matrix

SUBROUTINE IDENTIFICATION

A. Title

VAREQ

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

DAUX

FUNCTION

Function is to account for the central body and J_2 effects and to evaluate the second derivatives for the variational equations.

USAGE

A. Calling sequence

Call VAREQ

B. Input

1. CØMMØN

CMU	GM of Earth
TR2	Magnitude squared of the radius vector from the center of the Earth to the vehicle
TR3	Magnitude cubed of the above vector
TR5	Magnitude to the fifth power of the above vector
TLIST	Numerical integration working storage
PMAT	Matrix of position dependent effects in the variational equations
VMAT	Matrix of velocity dependent effects in the variational equations
FJ	Array containing the desired zonal harmonic constants (J_1, J_2, $\cdots$, J_{12})
NDPR	Number of Category 1 variables being solved for
NICPR	Number of initial condition (α, δ, β, A, R,v) parameter being solved for
NIDP	Start of array in IVSTR identifying the Category 1 variables being solved for
CDAD2M	Drag parameter $C_D A/2m$
TDØN	Drag variation modifier
CK	Drag variation K
TDRAG	Array containing the perturbative acceleration of the vehicle due to atmospheric drag in the x, y, and z directions

2. Calling sequence

—

C. Output

1. CØMMØN

TLIST Numerical integration working storage

2. Calling sequence

—

D. Error/action message

—

SUBROUTINES USED

A. Library

—

B. Program

ØUTER

EQUATIONS

A. Central body effects in PMAT

$$\mathrm{PMAT} = \mathrm{PMAT} + \begin{bmatrix} \frac{3\mu}{R^5} x^2 - \frac{\mu}{R^3} & \frac{3\mu}{R^5} xy & \frac{3\mu}{R^5} xz \\ \frac{3\mu}{R^5} xy & \frac{3\mu}{R^5} y^2 - \frac{\mu}{R^3} & \frac{3\mu}{R^5} yz \\ \frac{3\mu}{R^5} xz & \frac{3\mu}{R^5} yz & \frac{3\mu}{R^5} z^2 - \frac{\mu}{R^2} \end{bmatrix}$$

B. J_2 effects in PMAT

$$S = \frac{15}{2} J_2 \frac{\mu}{R^7} \left(1 - \frac{7z^2}{R^2}\right)$$

$$T = \frac{15}{2} J_2 \frac{\mu}{R^7} \left(3 - \frac{7z^2}{R^2}\right)$$

$$U = \frac{3}{2} J_2 \cdot \frac{\mu}{R^5} \left(1 - 5 \frac{z^2}{R^2}\right)$$

$$\text{PMAT} = \text{PMAT} + \begin{bmatrix} x^2S - U & xyS & xzT \\ xyS & y^2S - U & yzT \\ xzT & yzT & z^2T - 3U \end{bmatrix}$$

$$\frac{d^2}{dt^2}\left(\frac{\partial \vec{x}}{\partial p_i}\right) = \text{PMAT}\left(\frac{\partial \vec{x}}{\partial p_i}\right) + \text{VMAT}\left(\frac{\partial \dot{\vec{x}}}{\partial p_i}\right) \qquad i = 1,\ 2,\ \cdots,\ \text{NDPR}$$

where

$$\frac{\partial \vec{x}}{\partial p_i} = \left[\frac{\partial x}{\partial p_i},\ \frac{\partial y}{\partial p_i},\ \frac{\partial z}{\partial p_i}\right]$$

C. When $C_D A/2m$ is a solution parameter

$$\frac{d^2}{dt^2}\left[\frac{\partial \vec{x}}{\partial\left(\frac{C_D A}{2m}\right)}\right] = \frac{d^2}{dt^2}\left[\frac{\partial \vec{x}}{\partial\left(\frac{C_D A}{2m}\right)}\right] + \frac{\ddot{\vec{x}}_{drag}}{\left(\frac{C_D A}{2m} + \text{TD}\emptyset\text{N} \cdot K\right)}$$

D. When K is a solution parameter

$$\frac{d^2}{dt^2}\left[\frac{\partial \vec{x}}{\partial (K)}\right] = \frac{d^2}{dt^2}\left[\frac{\partial \vec{x}}{\partial (K)}\right] + \frac{\ddot{\vec{x}}_{drag}}{\left(\frac{C_D A}{2m} + \text{TD}\emptyset\text{N} \cdot K\right)} \cdot \text{TD}\emptyset\text{N}$$

SUBROUTINE IDENTIFICATION

A. Title

VPERT

B. Segment

ESPØDDC

ESPØDEPH

C. Called by subroutines

SETIC (ESPØDDC, ESPØDEPH)

FUNCTION

Function is to compute the partials of the Cartesian coordinates with respect to desired Category 1 parameters and to initialize the integration list with these partials.

USAGE

A. Calling sequence

Call VPERT

B. Input

1. CØMMØN

TICRT	Nominal Cartesian coordinates
TIPØL	Nominal spherical coordinates
IVSTR	Fixed point variable storage
NDPR	Total number of Category 1 variables to solve for
NIDP	Identifier for table indicating Category 1 type variables to be solved for
TEMP	Temporary storage

2. Calling sequence

—

C. Output

1. CØMMØN

TLIST Numerical integration working storage

2. Calling sequence

—

D. Error/action messages

SUBROUTINES USED

A. Library

CØSF

SINF

B. Program

—

EQUATIONS

Initialize variational equations.

α (right ascension)

$$\left.\begin{array}{ll} \dfrac{\partial x}{\partial \alpha} = -y & \dfrac{\partial \dot{x}}{\partial \alpha} = -\dot{y} \\[2ex] \dfrac{\partial y}{\partial \alpha} = x & \dfrac{\partial \dot{y}}{\partial \alpha} = \dot{x} \\[2ex] \dfrac{\partial z}{\partial \alpha} = 0 & \dfrac{\partial \dot{z}}{\partial \alpha} = \dot{0} \end{array}\right\} P_1$$

δ (declination)

$$\left.\begin{array}{ll} \dfrac{\partial x}{\partial \delta} = -r \sin \delta \cos \alpha & \dfrac{\partial \dot{x}}{\partial \delta} = \dot{z} \cos \alpha \\[2ex] \dfrac{\partial y}{\partial \delta} = -r \sin \delta \sin \alpha & \dfrac{\partial \dot{y}}{\partial \delta} = -\dot{z} \sin \alpha \\[2ex] \dfrac{\partial z}{\partial \delta} = r \cos \delta & \dfrac{\partial \dot{z}}{\partial \delta} = v\,(\cos \beta \cos \delta \\ & \qquad - \cos A \sin \beta \sin \delta) \end{array}\right\} P_2$$

β (flight path angle)

$$\left.\begin{array}{l} \dfrac{\partial x}{\partial \beta} = \dfrac{\partial y}{\partial \beta} = \dfrac{\partial z}{\partial \beta} = 0 \\[2ex] \dfrac{\partial \dot{x}}{\partial \beta} = -v\left[(\sin \beta \cos \delta + \cos A \cos \beta \sin \delta) \cos \alpha + \sin A \cos \beta \sin \alpha\right] \\[2ex] \dfrac{\partial \dot{y}}{\partial \beta} = -v\left[(\sin \beta \cos \delta + \cos A \cos \beta \sin \delta) \sin \alpha + \sin A \cos \beta \cos \alpha\right] \\[2ex] \dfrac{\partial \dot{z}}{\partial \beta} = v(\cos A \cos \beta \cos \delta - \sin \beta \sin \delta) \end{array}\right\} P_3$$

A (azimuth)

$$\left.\begin{aligned}
&\frac{\partial x}{\partial A} = \frac{\partial y}{\partial A} = \frac{\partial z}{\partial A} = 0 \\
&\frac{\partial \dot{x}}{\partial A} = v(\sin A \sin \delta \cos \alpha - \cos A \sin \alpha) \sin \beta \\
&\frac{\partial \dot{y}}{\partial A} = v(\sin A \sin \delta \sin \alpha + \cos A \cos \alpha) \sin \beta \\
&\frac{\partial \dot{z}}{\partial A} = -v(\sin A \cos \delta \sin \beta)
\end{aligned}\right\} P_4$$

r (magnitude of radial vector)

$$\left.\begin{aligned}
&\frac{\partial x}{\partial r} = \frac{x}{r} \\
&\frac{\partial y}{\partial r} = \frac{y}{r} \\
&\frac{\partial z}{\partial r} = \frac{z}{r} \\
&\frac{\partial \dot{x}}{\partial r} = \frac{\partial \dot{y}}{\partial r} = \frac{\partial \dot{z}}{\partial r} = 0
\end{aligned}\right\} P_5$$

v (velocity)

$$\left.\begin{aligned}
&\frac{\partial x}{\partial v} = \frac{\partial y}{\partial v} = \frac{\partial z}{\partial v} = 0 \\
&\frac{\partial \dot{x}}{\partial v} = \frac{\dot{x}}{v} \\
&\frac{\partial \dot{y}}{\partial v} = \frac{\dot{y}}{v} \\
&\frac{\partial \dot{z}}{\partial v} = \frac{\dot{z}}{v}
\end{aligned}\right\} P_6$$

SUBROUTINE IDENTIFICATION

A. Title

WEØFT

B. Segment

ESPØD

ESPØDDC

C. Called by subroutine

LØDØBS	(ESPØD)
WRTCØM	(ESPØD, ESPØDDC)

FUNCTION

Function is to write a sentinel block on the program work tape.

USAGE

A. Calling sequence

Call WEØFT

B. Input

1. CØMMØN

DBUFS	Auxiliary buffer storage
MT	Observation tape number

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

WRTCØM

B. Segment

ESPØD

ESPØDDC

C. Called by subroutines

DPRØS	(ESPØD)
FIT	(ESPØDDC)
INTEG	(ESPØDDC)
MAIN CONTROL	(ESPØD)

FUNCTION

Function is to write CØMMØN data storage on the work tape. The routine writes a fixed number of blocks, on the work tape "MT," of consecutive cells from the start to the end of CØMMØN. The first 128 word block written on tape contains blanks except for the first and second words. "70 Δ TAPE 7" is the first word. "OXXXΔΔΔΔ" is the second word, where the XXX represent the vehicle number. Δ represents blank. A sentinel block is written on the tape after all the data is written.

USAGE

A. Calling sequence

Call WRTCØM

B. Input

1. CØMMØN

CWE	Earth's rotational rate
DBUFS	Auxiliary buffer storage
DVEHN	Vehicle number and name (BCD)
MT	Observation tape number

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINES USED

A. Library

MXØRD

B. Program

REWT	Rewinds the observation tape
WEØFT	Writes an E.Ø.F. on the observation tape

SUBROUTINE IDENTIFICATION

A. Title
WRTØBS

B. Segment
ESPØD

C. Called by subroutine
LØDØBS

FUNCTION

Function is to write observations on observation tape in blocks. The sets are written as full 128 word blocks with a maximum of 55 blocks (704 observation sets) written at any one entry into the subroutine.

USAGE

A. Calling sequence
Call WRTØBS

B. Input

1. CØMMØN

MT	Observation tape number
NMBER	Number of observations
TEMP	Temporary storage

2. Calling sequence

—

C. Output

1. CØMMØN

—

2. Calling sequence

—

D. Error/action messages

—

SUBROUTINE IDENTIFICATION

A. Title
XCRØSS

B. Segment
ESPØDDC
ESPØDEPH

C. Called by subroutines

PARØUT	(ESPØDDC)
PRAVIS	(ESPØDEPH)

FUNCTION

This subroutine performs the cross product of two three-dimensional vectors.

$C = A \times B$

USAGE

A. Calling sequence
Call XCRØSS (A, B, C)

B. Input
1. CØMMØN
—
2. Calling sequence

A	Three-dimentional vector
B	Three-dimentional vector

C. Output
1. CØMMØN
—
2. Calling sequence

C	Three-dimentional solution vector

D. Error/action messages
—

XCRØSS

SUBROUTINES USED

A. Library

—

B. Program

—

SUBROUTINE IDENTIFICATION

A. Title

YHADEC

B. Segment

ESPØDDC

C. Called by subroutine

PARØUT

FUNCTION

Function is compute the vector Y, when range, hour angle and declination are given. The range measurement is assumed to be the computed range since it is not a measurement reported with hour angle and declination.

USAGE

A. Calling sequence

Call YHADEC (R, Y)

B. Input

1. CØMMØN

PSTAT(7)	w_1^s	coordinates of the sensor in the w system
PSTAT(8)	w_3^s	
PUBS(7)	Measured hour angle	
PUBS(8)	Measured declination	

2. Calling sequence

R Computed slant range

C. Output

1. CØMMØN

2. Calling sequence

Y Vector Y (y_1, y_2, y_3)

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF

SINF

B. Program

—

EQUATIONS

$$\vec{y} - \begin{bmatrix} w_1^s \\ 0 \\ w_3^s \end{bmatrix} + (R \cos H \cos \delta) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - (R \sin H \cos \delta) \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + (R \sin \delta) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

where H and S are the measured hour angle and declination measurements

SUBROUTINE IDENTIFICATION

A. Title

YRAE

B. Segment

ESPØDDC

C. Called by subroutine

PARØUT

FUNCTION

Function is to compute the vector Y, when range azimuth, and elevation are given. The Y vector represents the measured position in the inertial reference frame. It is used in the computation of the up, down, cross residuals.

USAGE

A. Calling sequence

Call YRAE(R, Y)

B. Input

1. CØMMØN

PSTAT(4)	$\cos \phi^*$	coordinates of the sensor in the w system
(5)	$\sin \phi^*$	
(7)	w_1^s	
(8)	w_3^s	

PUBS (4)	Azimuth (measured)
(5)	Elevation (measured)

PCSA	cosine of the computed azimuth
PCSE	cosine of the computed elevation
PSNA	sine of the computed azimuth
PSNE	sine of the computed elevation

2. Calling sequence

R Slant range (this will be the measured range if it is available. Otherwise it is the computed range.)

C. Output

1. CØMMØN

2. Calling sequence

Y Vector Y (y_1, y_2, y_3)

YRAE

D. Error/action messages

—

SUBROUTINES USED

A. Library

CØSF

SINF

B. Program

—

EQUATIONS

$$\vec{y} = \begin{bmatrix} W_1^s \\ 0 \\ W_3^s \end{bmatrix} + (R \sin A \cos E)\,\overrightarrow{EAST} + (R \cos A \cos E)\,\overrightarrow{NØRTH} + (R \sin E)\,\overrightarrow{VERT}$$

where

$$\overrightarrow{EAST} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \qquad \overrightarrow{NORTH} = \begin{bmatrix} -\sin \phi^* \\ 0 \\ \cos \phi^* \end{bmatrix} \qquad \overrightarrow{VERT} = \begin{bmatrix} \cos \phi^* \\ 0 \\ \sin \phi^* \end{bmatrix}$$

and R, A, E represent the measured observations. In the absence of any one of these three, the computed observation will be used.

5. OPTIONS AND PROCESSES

This section presents the background information and general organization of the principal options, processes, and interpretations of the ESPOD program. The rationale of the program parts and the analyst oriented discussions of the processes are given. This section supports the ESPØD Operating Instructions and Card Formats report, ESD-TDR-64-394. The operating instructions have brief descriptions of the necessary input options to identify the necessary constants and flags to control input. This section gives information needed by the analyst to structure the inputs to accomplish special tasks and to interpret the slight variations of certain outputs.

5.1 INITIAL CONDITIONS

Initial conditions are an estimate of the orbit elements of the satellite which are available in advance. These conditions give the position and velocity of the satellite for an arbitrary epoch, in an arbitrary coordinate system, and in an arbitrary format. ESPOD accepts five different kinds of initial conditions in three different formats. It updates these to epochs selected under any of four sets of rules. The analyst controls precisely the option he obtains by his choice of preliminary data input (or absence of input). Table 5-I presents the available options; details on the use of the input cards to select these options are given in Section 2.1.2.1 of ESD-TDR-64-394, ESPØD Operational Instructions and Card Formats.

It is mandatory that initial conditions of some type be provided to ESPOD to initiate the integration of the orbit, and that these initial conditions be established for an epoch within or relatively close to the interval of time spanned by the observations. Usually the conditions will be specified in the form of SPADATS mean elements at the epoch for which they were established. This epoch will not fall typically at a time useful to the current set of data and must be updated (or back dated) to a more appropriate time. Updating to the time of the last observation of the current set is automatic for elements read from the SEAI tape unless some other epoch is specified on a DNREV card.

The DNREV card allows the analyst to specify the required epoch as a certain number of revolutions (integer plus fraction) either from the first

Table 5-I. Initial Condition and Epoch Options

Initial Conditions Type (ICTYP)	1	2	3	4a	4b	Not Specified (a)	Not Specified (b)
Initial Conditions	αδβARv	λδβARv	$x, y, z, \dot{x}, \dot{y}, \dot{z}$	SPADATS Mean Elements (1963)	SPADATS Mean Elements (1964-K25)	SPADATS Mean Elements (1963)	SPADATS Mean Elements (1964-K25)
FORMAT	ICØND Cards	ICØND Cards	ICØND Cards	SPADATS 7-card set	SPADATS 7-card set	SEAI Tape	SEAI Tape
EPOCH SPECIFICATION							
NONE	NA Input Error	NA Input Error	NA Input Error	EPOCH given with 7-card set	EPOCH given with 7-card set	ESPØDDC: Time of last observation (ESPØDEPH only: File epoch)	ESPØDDC: Time of last observation (ESPØDEPH only: File epoch)
ITIME CARD	Epoch at time specified on ITIME card	Epoch at time specified on ITIME card	Epoch at time specified on ITIME card	NA Ignored	NA Ignored	NA Ignored	NA Ignored
DNREV CARD Specify: 1) Days since 1 January 2) Revs since epoch given with SPADATS mean elements 3) Revs since ascending node following launch 4) Time of last observation	NA Ignored	NA Ignored	NA Ignored	Update elements to epoch implied on DNREV card	Update elements to epoch implied on DNREV card	Update elements to epoch implied on DNREV card	Update elements to epoch implied on DNREV card

NA = Not applicable

ascending node after launch or from the node at which the SPADATS elements are given. It also allows the analyst to specify a time for epoch, given as day number (integer plus fraction) with 0.0^h, January 1 equal to 1.0 or to indicate that the time of the last observation will be taken as epoch.

ESPOD also accepts osculating elements at a given epoch as initial conditions. The input card format for osculating elements is identical to ESPØD output card format; hence, the elements resulting from any ESPOD iteration may be selected and used as initial conditions for further processing of the same case. In order to update (or backdate) osculating elements to a different epoch, it is necessary to integrate the orbit with the ESPØDEPH module. The given osculating elements at the present epoch are provided to ESPØDEPH as initial conditions with the difference of time between the present and required epoch specified as the update interval on a DELTT card. Osculating elements updated to the new epoch are printed on the output listing and may be transcribed to ICØND cards for subsequent input.

5.2 SOLUTION VECTOR

5.2.1 Limiting Dimension of Solution Vector

ESPOD calculates the differential correction for any number of variables between 1 and 50 depending on the allocation of variable core storage. Considerations regarding the allocation of variable storage, as explained in Section 3.1.2, lead to the following inequality

$$n \leq \sqrt{3072.028 - \frac{2D + 28\,S - m}{3}} - 5.167$$

or approximately

$$n \leq 50.26 - \frac{2D + 28\,S - m}{332.6}$$

where n is the dimension of the solution vector, D is the number of pairs of residual identifiers entered on DELET cards ($D \leq 25$), S is the number of sensors contributing data, and m is the dimension of the Category 1 portion of the solution vector.

NOTE This formula applies only if no *a priori* S matrix is used (the normal case).

EXAMPLE

D = 0, no items manually deleted

S = 10, ten contributing sensors

m = 8, all Category 1 variables in use

n $\leq$ 49.43, that is, the solution vector may be of any dimension up to 49. 8 dimensions are occupied with Category 1 variables leaving 41 for Category 2 variables, or approximately 4 per sensor.

5.2.2 Specification of Solution Vector

The precise complement included in the solution vector (see Table 5-I for all possibilities) is defined by inputs on the CAT1 and CAT2 cards. Category 1 variables are concerned with the motion of the satellite. Initial or starting estimates of these variables must be provided to initiate the differential correction. Variables 1 to 6 are derived automatically from the initial conditions which describe the orbit. Variables 7 and 8 must be provided as input on the DRAG card, but even lacking any estimate whatsoever, they must be set initially to some nonzero value. Category 2 variables are concerned with the biases associated with the sensors. Generally, in using ESPOD to solve for the orbit elements of a satellite, Category 2 variables, when included provide compensation for suspected biases in sensors. They are not in such a case typically included to calibrate the sensors. However, they are appropriate for use in sensor calibration when particular care is exercised in choosing the object satellite. It is not required to provide initial estimates of the biases when solving for Category 2 variables, but it will usually hasten convergence if they are approximately known. Initial estimates of biases are input with BISES cards with the preliminary data.

5.2.3 Bounds (See also Section 5.6.3)

The differential correction process changes each variable in the solution vector by an amount calculated to minimize the root mean square weighted residuals (see Section 5.4 for definition of residuals). These calculated amounts are, however, subjected to upper limits by BOUNDS incorporated as side conditions in the solution process. Bounds have the principal function of limiting the differential corrections to a value permitted by

the linear approximation to the nonlinear convergence process (see Section 5.6). Bounds may also be set to zero and thus used to hold constant a variable contained in the solution vector. A Category 1 variable may be held constant by omitting it from the solution vector, or by setting its corresponding bound to zero; its effect as an initial condition is in any case retained for integrating the orbit. A sensor bias, if it is known to exist as a constant, may be selected as a variable from Category 2, included in the solution vector, entered as an estimated bias on the BISES card, and then held CONSTANT by setting the corresponding bound to zero. Note that if any bound is changed with a BNDS card, all bounds must be presented and specified (see Section 2.1.3.2.4 of ESD-TDR-64-394).

The selection of initial bounds is a matter requiring experience on the part of the analyst. (Nominal values applicable for near Earth near circular orbits have been built into the program.) This typical set is shown in Table 5-II. The bounds for any particular case would depend upon the case, that is, upon the quality of the data, the dimension of the solution vector, the complement of the solution vector, the stability of the solution vector, the accuracy to which initial biases are known, the correlations between variables, whether the normal matrix is ill or well-conditioned, the quality of the initial conditions, and the set of contributing sensors. No rule can be given for choosing optimum bounds because their effect is apparent only in already difficult convergence problems. Experience indicates that analysts often err at first by choosing bounds too tight.

Table 5-II. Possible Elements in Solution Vector

Solution Element Class	Variable No.	Symbol	Definition	Specimen (Bounds and Units)
Category I Variables				
Osculating polar orbital elements at epoch	1	α	Right ascension	1 deg
	2	δ	Declination	1 deg
	3	β	Flight path angle	0.2 deg
	4	A	Inertial azimuth	0.5 deg
	5	R	Range	1 km
	6	V	Velocity	0.1 km/sec
Drag Parameters	7	$C_DA/2m$	Drag parameter	1 unit $C_DA/2m$
	8	K	Variation in drag parameter	0.1 unit $C_DA/2m$
Category II Variables				
Bias in observations from an arbitrary sensor S	9	R_b	Range bias	10 km
	10	A_b	Azimuth bias	5 deg
	11	E_b	Elevation bias	5 deg
	12	R_b	Range rate bias	0.5 km/sec
	13	α_b	Right ascension bias	5 deg
	14	δ_b	Declination bias	5 deg
	15	T_b	Time bias	10 sec
Bias in location of an arbitrary sensor S	16	ϕ_b	Latitude bias	0.1 deg
	17	λ_b	Longitude bias	0.1 deg
	18	h_b	Height bias above ellipsoid	0.1 km

5.3 SENSOR DATA

Sensor data in ESPOD are assigned to observations in the ESPØD segment of the program (see Table 5-III). The identifying number of the observing SPADATS sensor is provided with the observation. The observations are scanned and all contributing sensors are identified. (The data for these sensors are then read from SEAI tape and stored in core.) Any sensor cards input with the preliminary data modify or supplement the data in core. Calculations are made to complete the Master Sensor Table. Every contributing sensor requires an entry in the STYPE table to specify its standard deviation class. This table is stored in ESPOD. Any STYPE cards input with the preliminary data modify or supplement the STYPE table in core. IF A CONTRIBUTING SENSOR IS NOT REPRESENTED IN THE STYPE TABLE, ITS OBSERVATIONS WILL NOT ENTER THE DIFFERENTIAL CORRECTION.

5.3.1 Source of Standard Deviations

Every sensor is assigned to a particular standard deviation class, except that FPS 49 sensors have their standard deviations computed as rational functions of credence reported on the observation card (Column 10). This standard deviation class is identified by the value called SIGMA type. The standard deviations proper are stored in the SIGMA table. The SIGMA type designates a particular row in the SIGMA table. Each row of the SIGMA table contains four values. The values are interpreted differently for radar measurements than for camera measurements.

Sensor	Value 1	Value 2	Value 3	Value 4
Radar	σ_R	σ_A	σ_E	$\sigma_{\dot{R}}$
BAKER-NUNN Camera	σ_α (FR)	σ_δ(FR)	σ_α(PR)	σ_δ(PR)

(FR) - Field Reduced Data
(PR) - Precision Reduced Data

If any standard deviation is entered as zero, the corresponding observation will not enter the differential correction, but the residual will be printed (without flags). The weighted square residual will not

Table 5-III. General Sensor Data

Item	Source	Alternate Source	ESPØD Storage Location	Use
SPADATS sensor number	Observation on SRADU tape	Observation Card	Observation tape LØG 7	Argument to enter STYPE table and master sensor tables
The following are cataloged by sensor number				
Latitude, ϕ (radians)	Sensor file on SEAI tape	Sensor Card	Master sensor table	
Longitude, λ (radians)	Sensor file on SEAI tape	Sensor Card	Master sensor table	
Height above ellipsoid (h/e.r.)	Sensor file on SEAI tape	Sensor Card	Master sensor table	
cos ϕ*	Calculated		Master sensor table	
sin ϕ*	Calculated		Master sensor table	
Sensor RA at 0^h on day of epoch, $\alpha_{g_o} + \lambda$ (radians)	Calculated		Master sensor table	
Distance to equatorial plane, W_3^S (e.r.)	Calculated		Master sensor table	
Distance to Earth axis, W_1^S (e.r.)	Calculated		Master sensor table	
Standard deviation class (sigma type)	Stored in ESPØD (STYPE Table)	STYPE Card	99 Table, Items 485-592	Argument to enter SIGMA table
FPS 49 indicator flag	Stored in ESPØD (STYPE Table)		99 Table, Items 485-592	Indicates standard deviations are functions of credence
Gross outlier criterion, G_s	Stored in ESPØD (STYPE Table)	STYPE Card	99 Table, Items 485-592	Gross outlier rejection editing
Refraction correction flag	Stored in ESPØD (STYPE Table)	STYPE Card	99 Table, Items 485-592	Indicates refraction correction is to be performed
Mean surface value of refractivity $\bar{N}_s$	Stored in ESPØD (STYPE Table)	STYPE Card	99 Table, Items 485-592	
Latitude bias, ϕ_b	BISES Card	Solved in DC	Solution Vector	$\phi + \phi_b$ = Effective latitude
Longitude bias, λ_b	BISES Card	Solved in DC	Solution Vector	$\lambda + \lambda_b$ = Effective longitude
Height bias, h_b	BISES Card	Solved in DC	Solution Vector	$h + h_b$ = Effective height above ellipsoid
Range bias, R_b	BISES Card	Solved in DC	Solution Vector	$R + R_b$ = Effective measured range
Azimuth bias, A_b	BISES Card	Solved in DC	Solution Vector	$A + A_b$ = Effective measured azimuth
Elevation bias, E_b	BISES Card	Solved in DC	Solution Vector	$E + E_b$ = Effective measured elevation
Range rate bias, $\dot{R}_b$	BISES Card	Solved in DC	Solution Vector	$\dot{R} + \dot{R}_b$ = Effective measured range rate
Right ascension bias, α_b	BISES Card	Solved in DC	Solution Vector	$\alpha + \alpha_b$ = Effective measured right ascension
Declination bias, δ_b	BISES Card	Solved in DC	Solution Vector	$\delta + \delta$ = Effective measured declination
Time bias, t_b	BISES Card	Solved in DC	Solution Vector	$t + t_b$ = Effective time of observation
The following are cataloged by sigma type				
Standard deviation in R, σ_R (km)	Stored in ESPØD (SIGMA Table)	SIGMA Card	99 Table, Items 225-334	Reciprocal of weight applied to range
Standard deviation in A, σ_A (deg)	Stored in ESPØD (SIGMA Table)	SIGMA Card		Reciprocal of weight applied to azimuth
Standard deviation in E, σ_E (deg)	Stored in ESPØD (SIGMA Table)	SIGMA Card		Reciprocal of weight applied to elevation
Standard deviation in $\dot{R}$, $\sigma_{\dot{R}}$ (m/sec)	Stored in ESPØD (SIGMA Table)	SIGMA Card		Reciprocal of weight applied to range rate
Standard deviation in α, σ_α (deg)	Stored in ESPØD (SIGMA Table)	SIGMA Card		Reciprocal of weight applied to right ascension
Standard deviation in δ, σ_δ (deg)	Stored in ESPØD (SIGMA Table)	SIGMA Card		Reciprocal of weight applied to declination

be included in the summary table of means and root mean squares by sensor and observation type following the residuals print. The SIGMA table is stored in the ESPØD segment. Any SIGMA cards input with the preliminary data modify or supplement the SIGMA table in core. This process is outlined in Table 5-IV.

Table 5-IV. Selection of Standard Deviations

Given	Observation Card
Data Punched	Sensor Number (and other special data identifier)
Refer to	Table of Sensor Data
Obtain	Sigma Type, $1 < n < 60$
Refer to	SIGMA TABLE, line n
Obtain	σ_R, σ_A, σ_E, $\sigma_{\dot{R}}$ (or σ_α, σ_δ) from line n of SIGMA table

5.3.2 Bias Removal

The latitude, longitude, height above the ellipsoid, and the time of observation are used to relate the position of the sensor to the inertial coordinate system of integration. If it is desired, apparent biases in these values may be solved for and added to the tabular values in order to reduce the net residuals. The range, azimuth (or right ascension), elevation (or declination), and range rate are used to relate the position and velocity of the spacecraft to the sensor. If it is desired, apparent biases in these values may be solved for and added to the observed values in order to reduce the residuals. The method for calling for these bias solutions is described in Section 5.2.

5.4 RESIDUALS, WEIGHTING, ROOT MEAN SQUARES, AND EDITING

5.4.1 Residuals

Residuals are defined in ESPOD as the

> "Actual observed value corrected for atmospheric refraction as required, from a given sensor at a given time"
>
> minus
>
> "Computed observed value as if spacecraft were on the trajectory specified by the current elements with the current perturbations and were seen from the same sensor at the same time"

or more briefly

"observed value" minus "computed value"

Residuals are calculated and printed as applicable for the principal observed values which are directly reported on observation cards (see Table 5-V).

Table 5-V. Notation for Reported Observations, Their Residuals and Standard Deviations

Variable	Observed	Computed	Residual	*A Priori* Standard Deviation for Sensor S
Range	R	R_i	ΔR	σ_{SR}
Azimuth	A	A_i	ΔA	σ_{SA}
Elevation	E	E_i	ΔE	σ_{SE}
Range rate	$\dot{R}$	$\dot{R}_i$	$\Delta\dot{R}$	$\sigma_{S\dot{R}}$
Right Ascension	α	α_i	$\Delta\alpha$	$\sigma_{S\alpha}$
Declination	δ	δ_i	$\Delta\delta$	$\sigma_{S\delta}$

Residuals are also calculated and printed out on option for other right-handed Cartesian coordinate systems for analyst convenience as listed below:

a) UVW — Orbit plane, one axis is line to earth center

b) STW — Orbit plane, one axis is tangent to instantaneous velocity

c) LLH — Topographic, latitude, longitude, height of sensor

These coordinate systems are defined in Section 7 of this volume.

5.4.2 Weighting

Residuals are weighted before inclusion in the differential correction process. Proper weighting causes more accurate measurements to have a larger effect on the differential correction than less accurate measurements. Weighting is accomplished by dividing residuals by the *a priori* standard deviation appropriate to the observation class and to

the sensor with which the observation was taken (see Section 5.3). Further, azimuth and declination residuals are weighted by cos E and cos α, respectively, to account for convergence of azimuth and declination lines at the zenith and poles respectively.

5.4.3 Editing Residuals

5.4.3.1 Preliminary Editing

ESPOD computes a variety of statistics under the general class of "Root Mean Square Weighted Residuals (RMSWR)." These differ according to the type of residuals included. They are developed for differing applications during the progress of the program. Observations or their residuals, or both, are subjected to a series of three tests before they are considered for inclusion in any RMSWR, and to a fourth test before inclusion in a differential correction. The first three tests are:

> Test 1: If the residual has been selected by the analyst to be omitted by being identified on a DELET preliminary input card, it is rejected. The residual is printed and tagged with an asterisk (*).
>
> Test 2: If the observation occurs at a time removed more than ten days from epoch, it is rejected. (Analyst may change the "10" to any other value with a TMAX card input with preliminary data.)
>
> Test 3: If the weighted residual exceeds a gross outlier criterion, it is rejected. (Analyst may modify the gross outlier criterion sensor-by-sensor by an STYPE card input with the preliminary data.) The residual is printed and tagged with a "G."

Test 1 provides an opportunity for an analyst to reject certain observations following a manual review of the residuals. Then, if he so desires, he may loosen the values associated with the gross outlier rejection and the KRMSWR rejection (described below); this will effectively disable any automatic rejection. This manual-deletion-only feature permits the least squares process to operate on the same set of residuals from one iteration to the next, thus leading the least squares process to a definite, unique solution. Test 2 provides an opportunity to reject an observation which has a mispunched time, and which would cause integration to run unnecessarily.

A gross outlier (Test 3) may arise from many causes, i.e., mis-punching, data handling, or transmission error. The gross outlier test is defined by a constant, G_s, maintained with the sensor data, and which may be changed with a STYPE card. G_s is multiplied by the <u>a priori</u> σ_R, σ_A, σ_E, $\sigma_{\dot{R}}$, σ_α, σ_δ as maintained in the designated line of the SIGMA table to obtain the upper bound within which ΔR, ΔA, ΔE, $\Delta \dot{R}$, $\Delta\alpha$, $\Delta\delta$, respectively, must lie. Typically G_s is chosen to result in approximate upper bounds of $\Delta R < 1000$ km, $\Delta A < 10^\circ$, $\Delta E < 10^\circ$, $\Delta \dot{R} < 10$ m/sec, but in no instance can G_s be greater than 9999. The analyst has the option to set any individual G_s to zero, thereby selectively eliminating data from a particular sensor.

5.4.3.2 <u>Definition of RMSWR for Editing Purposes</u>

Weighted residuals which pass these first three tests are squared and summed separately over all sensors by class of observation.

$$\mathrm{RMSWR}_R = \left[\frac{1}{N'_R} \sum \left(\frac{\Delta R_i}{\sigma_{S_i R}}\right)^2\right]^{1/2}$$

$$\mathrm{RMSWR}_A = \left[\frac{1}{N'_A} \sum \left(\frac{\Delta A_i \cos E_i}{\sigma_{S_i A}}\right)^2\right]^{1/2}$$

$$\mathrm{RMSWR}_E = \left[\frac{1}{N'_E} \sum \left(\frac{\Delta E_i}{\sigma_{S_i E}}\right)^2\right]^{1/2}$$

$$\mathrm{RMSWR}_{\dot{R}} = \left[\frac{1}{N'_{\dot{R}}} \sum \left(\frac{\Delta \dot{R}_i}{\sigma_{S_i \dot{R}}}\right)^2\right]^{1/2}$$

$$RMSWR_{\alpha} = \left[\frac{1}{N'_{\alpha}} \sum \left(\frac{\Delta\alpha_i \cos\alpha_i}{\sigma_{S_{i\alpha}}}\right)^2\right]^{1/2}$$

$$RMSWR_{\delta} = \left[\frac{1}{N'_{\delta}} \sum \left(\frac{\Delta\delta_i}{\sigma_{S_{i\delta}}}\right)^2\right]^{1/2}$$

This set of RMSWR's is developed on every iteration for use in editing residuals for inclusion in the differential correction process on the immediately following iteration. They are not printed.

5.4.3.3 Editing by KRMSWR

On every iteration except the first, K times the $RMSWR_{\phi}$ (where ϕ is a typical observation) as developed on the previous iteration is used as a further test for rejecting outlying residuals before they enter into any differential correction process or any other RMSWR's.

> Test 4: If the weighted residual exceeds 1.5 times the $RMSWR_{\phi}$ which was calculated on the previous iteration, it is rejected. (K nominally equals 1.5. An analyst may change K to any other value with a 99-card, item 452, with preliminary data input.) The residual is printed and tagged with a "K."

This test does not apply on the first iteration and never applies to the calculation of $RMSWR_{\phi}$ (see Figures 5-1 through 5-3).

5.4.4 Definition of RMSWR for Test of Fit

Residuals which pass the Test 4 enter into the sum total RMSWR used as the criterion for judging elements. It is this total RMSWR which is minimized in the differential correction process

$$RMSWR = \left\{\frac{1}{N}\left[\sum\left(\frac{\Delta R_i}{\sigma_{S_iR}}\right)^2 + \sum\left(\frac{\Delta A_i \cos E_i}{\sigma_{S_iA}}\right)^2 + \sum\left(\frac{\Delta E_i}{\sigma_{S_iE}}\right)^2 + \sum\left(\frac{\Delta\dot{R}_i}{\sigma_{S_i\dot{R}}}\right)^2 + \sum\left(\frac{\Delta\alpha_i \cos\delta_i}{\sigma_{S_i\alpha}}\right)^2 + \sum\left(\frac{\Delta\delta_i}{\sigma_{S_i\delta}}\right)^2\right]\right\}^{1/2}$$

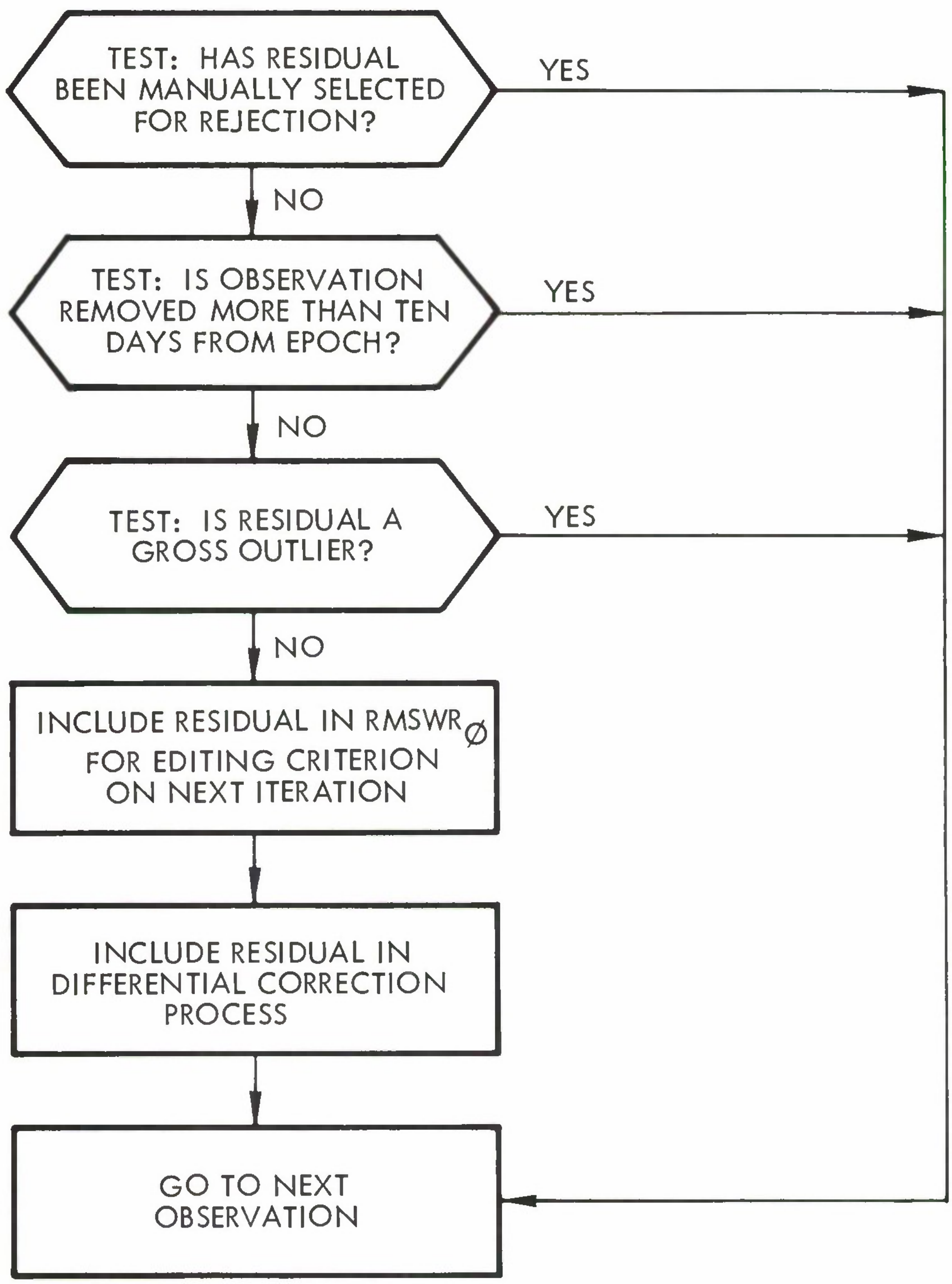


Figure 5-1. Editing Process on First Iteration

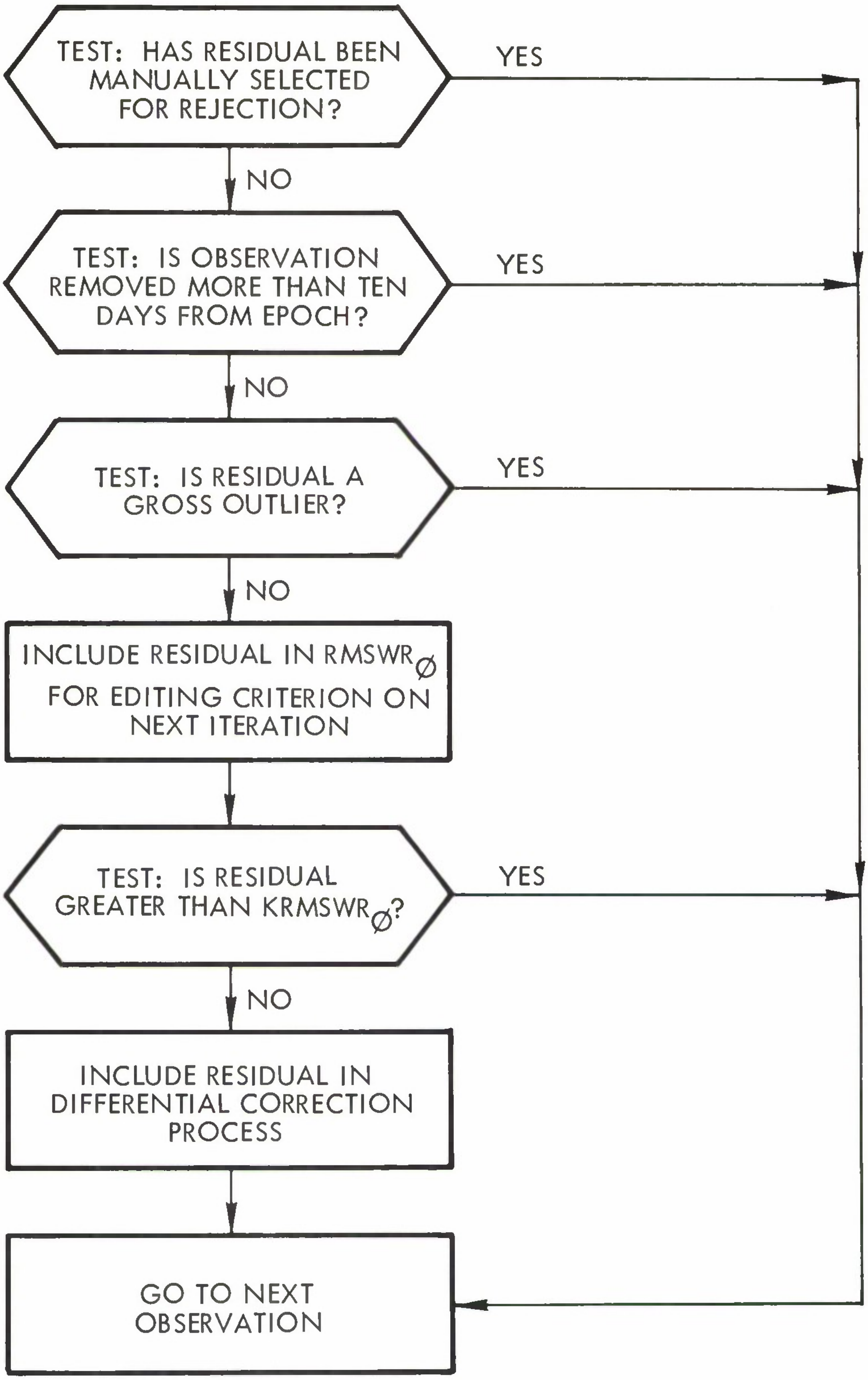
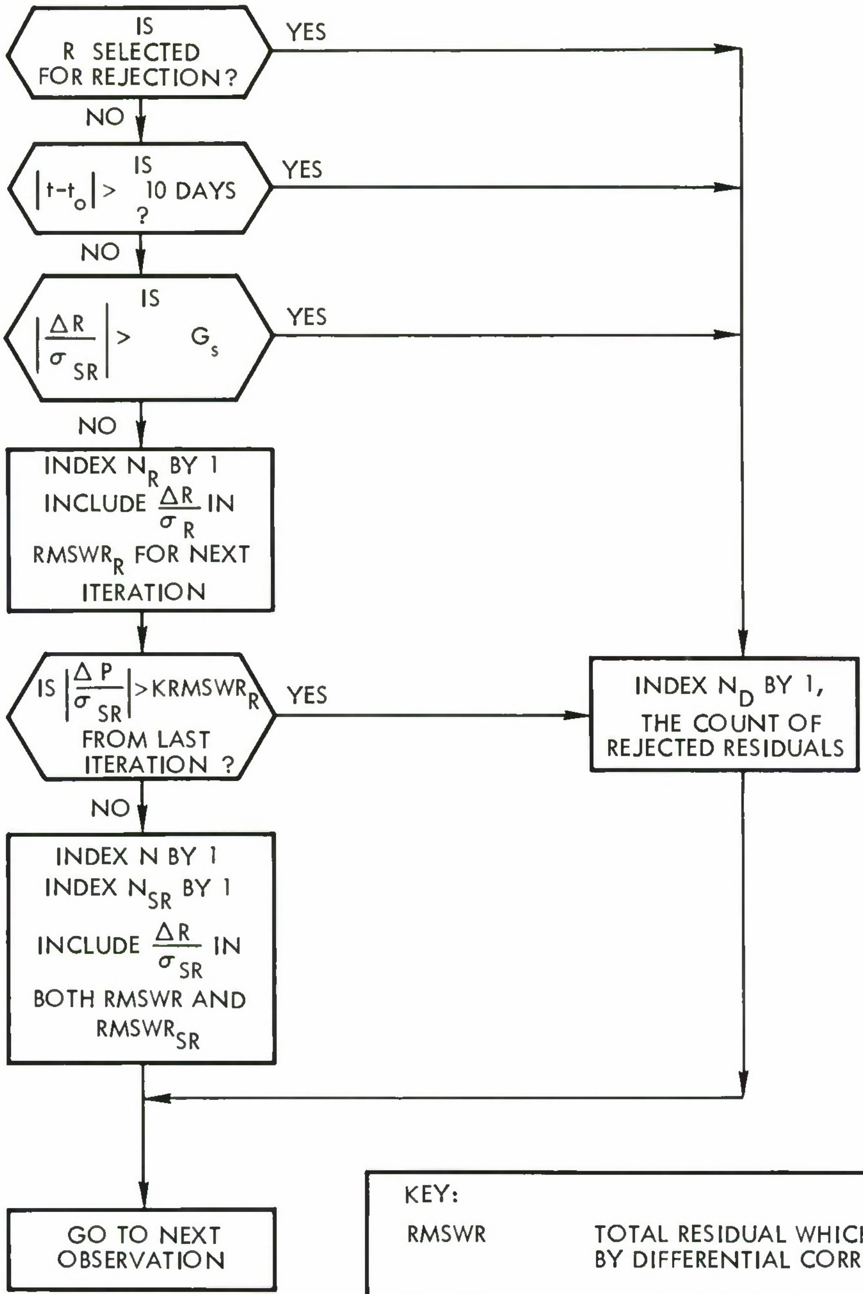


Figure 5-2. Editing Process on Iterations After the First



KEY:

RMSWR	TOTAL RESIDUAL WHICH IS MINIMIZED BY DIFFERENTIAL CORRECTION PROCESS
$RMSWR_R$	VALUE USED IN REJECTION CRITERION FOR RANGE RESIDUALS ON NEXT INTERATION
$RMSWR_{SR}$	MEASURE OF PERFORMANCE OF SENSOR S IN OBSERVING RANGE
N	NUMBER OF RESIDUALS ENTERING INTO RMSWR
N_R	NUMBER OF RESIDUALS ENTERING INTO $RMSWR_R$
N_{SR}	NUMBER OF RESIDUALS ENTERING INTO $RMSWR_{SR}$

Figure 5-3. Rejection Criteria for ΔR Taken at Sensor S on An Iteration Other Than the First

where N is the number of residuals passing all four tests. The RMSWR for the current iteration and the previous best RMSWR are printed following the residual summary at the end of each iteration.

In the interest of providing detailed information about the performance of each sensor, the RMSWR's are also accumulated by sensor and type of observation. These $RMSWR_{S\phi}$'s do not enter into any calculations, but are derived for information only. They are also printed following the residual summary at the end of each iteration.

$$RMSWR_{S\phi} = \left[\frac{1}{N_{S\phi}} \sum \left(\frac{\Delta\phi_i}{\sigma_{S\phi}}\right)^2\right]^{1/2}$$

5.5 REFRACTION CORRECTION

ESPOD will, on option, correct errors in observed elevation angles due to refraction in the troposphere. Nominally no correction is made. To scale the correction, a nonzero estimate for the mean surface value of refraction $Y_i(\overline{N}_s)$ is input on the STYPE card with preliminary data input. To call the correction into use, a refraction flag is input on the STYPE card.

If data from any one station is always corrected for refraction, ESPOD can be compiled to perform the correction automatically. A check may be made to determine the automatic mode by interpreting the sensor type table printed on option.

5.6 ESPOD CONVERGENCE LOGIC AND CONTROL

5.6.1 Process Control

ESPOD iterates on the elements provided as initial conditions to derive new elements which minimize the RMSWR of observations. This process is terminated according to various criteria controlled by the analyst or built into the program, or both. Possible conditions for termination are:

- Program determines that the next differential correction applied to the current "best elements to date" will not be sufficiently improved to change the RMS by as much as 0.1 percent (Analyst may change the "0.1 percent" to any other value by entering a 99 card item 00454 with preliminary data input.)

- Program has integrated the orbit five times without having obtained the convergence condition defined immediately above. Program terminates, saving on option either the last obtained approximate elements or the "best elements to date." (Analyst may change the "5" to any other value using a NITER card with the preliminary data input.)

- Program is unable to obtain a converging step from a set of starting elements even though the differential correction has been subjected to successively tighter bounds on four successive iterations. (The size of initial bounds is controlled by the analyst.)

The convergence problems that the analyst directly controls are:

- Determing that convergence has occurred
- Limiting the number of iterations performed
- Limiting the size of the differential correction in cases of an ill-conditioned normal matrix
- Rejecting truly impossible problems

5.6.2 Iteration Process

ESPOD proceeds one iteration at a time. The essential and time consuming part of the iteration is the integration of the orbit and the calculation of the residuals and partial derivatives. Once this is complete, the RMSWR has been calculated and compared with the best RMSWR previously obtained. If it is smaller, then the elements of the current orbit are the new "best elements to date" obtained and it is desirable to test whether further iterations are worth pursuing. This is accomplished by calculating the differential correction derived from this iteration, establishing new elements, and predicting by linear theory what the RMSWR due to them should be; this process does not require an integration of the orbit. If the predicted RMSWR is less than 0.1 percent of the current RMSWR, then the current best elements are defined as the solution, e.g., the process has converged. When this criterion is not satisfied, another iteration is required and the program proceeds again to integrate the orbit using the new elements and to test whether they are indeed better than the current best. This general iteration is presented in Table 5-VI steps 1 to 10. Step 11 permits the analyst to truncate the process after an arbitrary number of iterations.

5.6.3 Bounds

5.6.3.1 Definition of Bounds

ESPOD calculates the differential corrections by solving the weighted least squares problem under a side condition that the individual elements of the differential correction do not exceed some respective bound, and the sum of their squares weighted by the inverse bounds squared do not exceed unity. $\left[\sum \left(P_i/B_i\right)^2 \leq 1\right]$. On any individual step it is not known whether the side condition will influence the solution or not. In a real case, whether it has

or not is printed out with the residual summary. These bounds are entered initially with the BNDS cards, but are subject to automatic tightening or loosening according to builtin logic.

Table 5-VI. Convergence General Flow, General Case

1.	Given ${}_kE_n$	E = Elements of orbit n = Number of this set of elements ${}_kE_n$ = kth trial at corrected elements E_n derived from E_{n-1} ${}_kE_n$ may or may not be better than E_{n-1} E_n = Results from a successful iteration, i. e., best elements to date. E_1 = Initial Conditions
2.	Calculate ${}_kR_n$	Residuals of observations with respect to orbit determined from ${}_kE_n$
3.	Calculate ${}_kRMS_n$	Root mean square of weighted residuals ${}_kR_n$
4.	Test: ${}_kRMS_n < RMS_{n-1}$ yes: continuation for accepting E_n as new best elements to date → Step 5 no: continuation for proceeding to a new trial → Step 14	This tests whether ${}_kE_n$ is better than E_{n-1} RMS_o = Machine maximum value, i. e., E_1 is always best to date.
5.	Write E_n on Tape LØG7	Saves E_n and related parameters for use in next iteration, on an immediately following conditioned start, and for immediately following ESPØDEPH runs.

Table 5-VI. Convergence General Flow, General Case (Continued)

6. Calculate — Calculates the appropriate differential corrections to obtain ${}_1E_{n+1}$, ${}_2E_{n+1}$, etc.

${}_1\Delta E_n$ — ${}_1\Delta E_n$ is a function of current bounds, B.

${}_2\Delta E_n$ — ${}_2\Delta E_n$ is a function of B/2.

${}_3\Delta E_n$ — ${}_3\Delta E_n$ is a function of B/4.

${}_4\Delta E_n$ — ${}_4\Delta E_n$ is a function of B/8.

7. Set k = 1

At this point, we proceed to apply the differential corrections attempting to achieve a succeeding E_{n+1} which results in an RMS_{n+1} smaller than RMS_n. We are prepared to terminate the process if RMS_{n+1} is not significantly different from RMS_n to merit a new integration of the orbit (Step 10), or if the number of iterations already performed equals some preassigned number (Step 11).

8. Continuation for trial ${}_kE_{n+1}$

Calculate ${}_k\Delta E_{n+1} = E_n + {}_k\Delta E_n$

9. Calculate ${}_kPRMS_{n+1}$ — Calculates the predicted RMS_{n+1} assuming linear partial derivatives of residuals with respect to elements, and using ${}_k\Delta E_n$.

10. Test: $\dfrac{{}_kPRMS_{n+1} - RMS_n}{RMS_n} < \epsilon$

No	Yes
continuation for significant difference - Step 11	continuation for negligible difference - Step 12

This tests whether the predicted RMS_{n+1} is significantly different according to criterion ϵ from RMS_{n+1}. The value ϵ is nominally 0.001, but may be changed by analyst option. If the difference is significant, the program will try to continue (see Step 11) in order to determine whether RMS_{n+1} is less than RMS_n, that is, whether ${}_kE_{n+1}$ is indeed a better set of elements than E_n.

11. Test: NITER Completed

No	Yes
continuation for not completed go with ${}_kE_{n+1}$ to Step 1	continuation for completed → Step 12

NITER specifies the number of iterations the analyst wishes to allow the program to run. The value of NITER is nominally five but may be changed on input.

Table 5-VI. Convergence General Flow, General Case (Continued)

12. Test: Record E_{n+1}

yes no

This permits the analyst to record $_kE_{n+1}$ on tape LØG7 (see note to Step 5 above) if he chooses. Nominally, $_kE_{n+1}$ is a trial value and would not be recorded, but for reasons particular to the case at hand, using $_kE_{n+1}$ for subsequent updates and conditioned starts may be desirable. Option is controlled on JDC card.

13. Write E_{n+1} on tape LØG7

This concludes the direct flow through the convergence logic. Exit is to ESPØDEPH if called otherwise to the next JDC card.

14. Continuation for proceeding to a new trial

At this point, $_kE_n (n > 1)$ has proved to be a diverging step and E_{n-1} is still the best set of elements to date. Typically, $_kE_n = E_{n-1} + {}_k\Delta E_{n-1}$. The next trial elements available are typically $_{k+1}E_{n-1}$. This procedure applies if $k < 4$.

Test: $k < 4$?

yes no

continuation for k + 1. Go with $_{k+1}\Delta E_{n-1}$ to Step 8.

At this point $_4E_n$ has proved to be a diverging step and E_{n-1} is the best set of elements achievable with the current bounds. No further automatic continuation is provided; the remark "BOUNDS OVER EIGHT FAILED" is printed. If ESPØDEPH is called, E_{n-1} will be used. This concludes the direct flow through the convergence logic, with the note that convergence was not achieved. Exit is to ESPØDEPH if called, otherwise to the next JDC card.

Each time new "best elements to date" are identified, and the actual RMSWR is within 10 percent of what it was predicted to be, bounds are doubled to permit a larger, desirable correction. Each time new "best elements to date" are identified and the actual RMSWR is not within 10 percent of what it was predicted to be, the bounds are held at the values resulting in new best elements. The bounds applied at any iteration are printed on the residual summary, as to whether or not the bounds affected the solution.

5.6.2.2 Application of Bounds

Each time the actual RMSWR proves to be greater than the best RMSWR to date, a solution with tighter bounds is tried. This tightening of bounds is an attempt to limit the differential correction to a region about the best elements to date where the linear theory is appropriate. Four solutions with successively tighter bounds are tried in an effort to improve the RMSWR before the program concludes that a converging step cannot be obtained from the current best elements, at least with the initial bounds. This circumstance is tested for in Step 4 (answer "no"), anticipated in Step 6, and controlled in Step 14 of Table 5-VI. The details of the process are illustrated in Figure 5-4 for a case of proceeding from E to E_2 through potentially four trials. The notation corresponds to that of Table 5-VI. The figure is abstracted for compactness; the steps in any trial correspond in order to Steps 8, 9, 10, 11 (Step 1 implicit), 2, 3, and 4 of those of Table 5-VI.

5.6.3 Choice of Elements for use with ESPØDEPH

After the last iteration, stopping either due to Step 10 or due to Step 11, the choice remains whether to proceed with the best elements to date as proved by the test in Step 4, or to proceed with the new elements obtained from the immediately preceding differential correction. The residuals and RMSWR of these elements have not yet been calculated. The best proved elements to date are automatically selected by ESPØD unless specific instructions are provided by the analyst via the JDC card. A particular orbit may be well behaved and successive iterations may be predicted in advance as yielding converging corrections, thus permitting a bolder operating philosophy (see Step 12, Table 5-VI).

5.6.4 Nonconverging Termination

If the process is terminated on a NITER test (Step 11, when $K > 1$) the program is operating in a region where convergence is more difficult to obtain. The analyst may choose to proceed with an ESPØDEPH run with either the best elements to date or new untried elements. However, if he attempts to follow immediately with a conditional start, the program will reject the attempt. See discussion for conditioned start, Section 5.14.

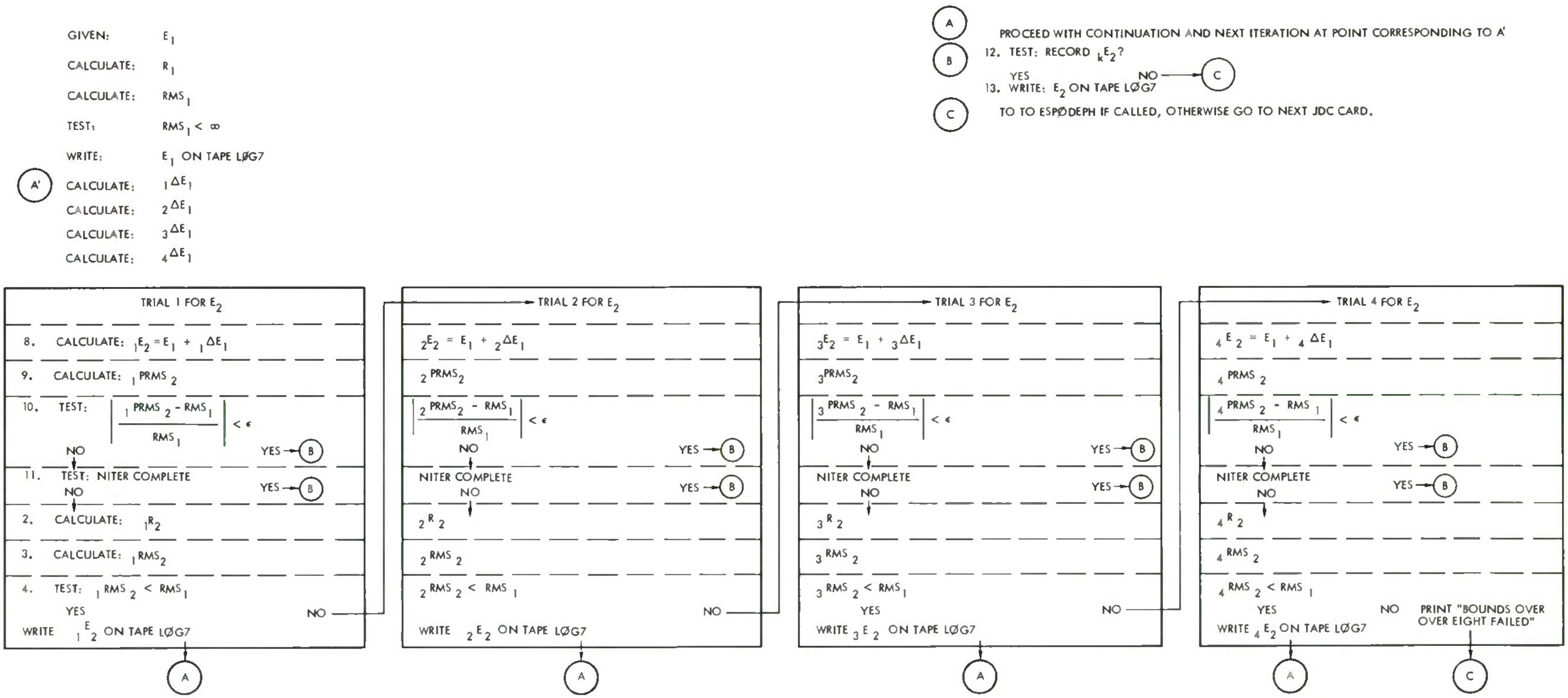


Figure 5-4. Convergence Logic — Abstract Flow Diagram for E_1 and Trial E_2's

5.7 INTEGRATION

The integration procedure used in ESPOD is general in nature and may be used to integrate any reasonably behaved system of second order ordinary differential equations of the form

$$y_i'' = f_i(t, y_1, \cdots, y_n\, y_1', \cdots, y_n'), \qquad i = 1, \cdots, n \tag{1}$$

In the present program Equations (1) consist of the equations of motion together with their associated variational equations. It will, however, be convenient to list the formulas as they apply to the general system (1) and thereby conserve on notation.

The method may be described as follows:

- Two numerical integrations take place simultaneously. One of these produces values of the functions y_i and the other produces values for their derivatives y_i'. The former is called the Cowell method and the latter, the Adams-Moulton method. Neither method is used in its simplest form. Rather, both are used in summed form in order to control the growth of round-off error. The Cowell method is summed twice and the Adams-Moulton once.

- Both the Cowell and the Adams-Moulton are multistep methods that require a certain number of starting values. In the present program these starting values are obtained using a single-step Runge-Kutta method.

5.7.1 Starting Method

The Runge-Kutta method is used to establish values of the functions y_i and their derivatives y_i' at eight successive time steps subsequent to epoch, t_o. At t_o, the values of y_i and y_i' are the initial position and velocity. Initially, a step-size Δt is provided. The step-size is nominally one minute. (Analyst may change this value to any other number of the form $P/2^n$, P integral, with a TSTEP card input with preliminary data.) Using the Runge-Kutta procedure the differential equations are integrated at a step-size $h = \Delta t/R$ until 8R integration steps have been completed. Nominally, R = 8. (Analyst may change the value 8 to another integral value with a 99-card Item 483 input with preliminary data.) Suppose that values $y_{i,j}$ and $y_{i,j}'$, for some $j < 8R$, of the functions y_i and their derivatives y_i', corresponding to the time $t_j = t_o + jh$, are known. To compute $y_{i,j+1}$ and $y_{i,j+1}'$ the following numbers are computed:

$$K_{i1} = hf_i\left(t_j,\ y_{i,j},\ y'_{i,j}\right) \tag{2}$$

$$K_{i2} = hf_i\left(t_j + \frac{h}{2},\ y_{i,j} + \frac{h}{2}y'_{i,j} + \frac{h}{8}K_{i1},\ y'_{i,j} + \frac{K_{i1}}{2}\right) \tag{3}$$

$$K_{i3} = hf_i\left(t_j + \frac{h}{2},\ y_{i,j} + \frac{h}{2}y'_{i,j} + \frac{h}{8}K_{i1},\ y'_{i,j} + \frac{K_{i2}}{2}\right) \tag{4}$$

$$K_{i4} = hf_i\left(t_j + h,\ y_{i,j} + hy'_{i,j} + \frac{h}{2}K_{i3},\ y'_{i,j} + K_{i3}\right) \tag{5}$$

The new values $y_{i,j+1}$ and $y'_{i,j+1}$ are then computed as

$$y_{i,j+1} = y_{i,j} + h\left[y'_{i,j} + \frac{1}{6}(K_{i1} + K_{i2} + K_{i3})\right] \tag{6}$$

$$y'_{i,j+1} = y'_{i,j} + \frac{1}{6}(K_{i1} + K_{i2} + K_{i3} + K_{i4}) \tag{7}$$

Thus, starting with $y_{i,0}$ and $y'_{i,0}$ as known quantities, the Runge-Kutta procedure, which consists of the computation outlined in Equations (2) through (7), repeats itself until $j + 1 = 8R$. At this point the values $y_{i,j}$ and $y'_{i,j}$ where $j = Rj'$, and $j' = 0, \cdots, 8$ are renumbered consecutively from 0 to 8 and the computation is continued as described in the next section.

5.7.2 Continuation Procedure

The values $y_{i,j}$ and $y'_{i,j}$, $j = 0, \cdots, 8$, are now used to compute a difference table (see Table 5-II).

The table to the right of the heavy vertical line is formed in the usual way. For that portion to the left of the vertical line, the following computations are made.

Table 5-VII. Difference Table

		$f_{i,0}$								
			$\nabla^1 f_{i,1}$							
		$f_{i,1}$		$\nabla^2 f_{i,2}$						
			$\nabla^1 f_{i,2}$		$\nabla^3 f_{i,3}$					
		$f_{i,2}$		$\nabla^2 f_{i,3}$		$\nabla^4 f_{i,4}$				
			$\nabla^1 f_{i,3}$		$\nabla^3 f_{i,4}$		$\nabla^5 f_{i,5}$			
		$f_{i,3}$		$\nabla^2 f_{i,4}$		$\nabla^4 f_{i,5}$		$\nabla^6 f_{i,6}$		
			$\nabla^1 f_{i,4}$		$\nabla^3 f_{i,5}$		$\nabla^5 f_{i,6}$		$\nabla^7 f_{i,7}$	
$''F_{i,3}$		$f_{i,4}$		$\nabla^2 f_{i,5}$		$\nabla^4 f_{i,6}$		$\nabla^6 f_{i,7}$		$\nabla^8 f_{i,8}$
	$'F_{i,4}$		$\nabla^1 f_{i,5}$		$\nabla^3 f_{i,6}$		$\nabla^5 f_{i,7}$		$\nabla^7 f_{i,8}$	
$''F_{i,4}$		$f_{i,5}$		$\nabla^2 f_{i,6}$		$\nabla^4 f_{i,7}$		$\nabla^6 f_{i,8}$		
	$'F_{i,5}$		$\nabla^1 f_{i,6}$		$\nabla^3 f_{i,7}$		$\nabla^5 f_{i,8}$			
$''F_{i,5}$		$f_{i,6}$		$\nabla^2 f_{i,7}$		$\nabla^4 f_{i,8}$				
	$'F_{i,6}$		$\nabla^1 f_{i,7}$		$\nabla^3 f_{i,8}$					
$''F_{i,6}$		$f_{i,7}$		$\nabla^2 f_{i,8}$						
	$'F_{i,7}$		$\nabla^1 f_{i,8}$							
$''F_{i,7}$		$f_{i,8}$								
	$'F_{i,8}$									
$''F_{i,8}$										

$$''F_{i,3} = \frac{y_{i,4}}{(\Delta t)^2} - C_0 f_{i,4} - C_2\nabla^2 f_{i,5} - C_4\nabla^4 f_{i,6} - C_6\nabla^6 f_{i,7} - C_8\nabla^8 f_{i,8}$$ †

$$'F_{i,4} = \frac{y'_{i,4}}{\Delta t} - D_0 f_{i,4} - D_1\nabla^1 f_{i,5} - D_2\nabla^2 f_{i,5} - D_3\nabla^3 f_{i,6}$$

$$- D_4\nabla^4 f_{i,6} - D_5\nabla^5 f_{i,7} - D_6\nabla^6 f_{i,7} - D_7\nabla^7 f_{i,8} - D_8\nabla^8 f_{i,8}$$

The difference table is then completed down to the diagonal line by the rule that the difference between any two consecutive entries in a vertical column is equal to the entry adjacent in the vertical column to the right. The computation then proceeds as follows:

The predicted values are computed as

$$\tilde{y}_{i,j+1} = (\Delta t)^2\left(''F_{i,j} + \sum_{r=0}^{8} \sigma_{r+2}\nabla^r f_{i,j}\right)$$

$$\tilde{y}'_{i,j+1} = (\Delta t)\left('F_{i,j} + \sum_{r=0}^{8} \gamma_{r+1}\nabla^r f_{i,j}\right)$$

Trial values of the next line of differences in the difference table are then formed, based on these quantities. That is, the differences $\nabla^r\tilde{f}_{i,j+1}$; $r = 0, \cdots, 8$, are computed and the final corrected values are then computed as

$$y_{i,j+1} = (\Delta t)^2\left(''F_{i,j} + \sum_{r=0}^{8} \sigma^*_{r+2}\nabla^r\tilde{f}_{i,j+1}\right)$$

$$y'_{i,j+1} = (\Delta t)\left('F_{i,j} + \sum_{r=0}^{8} \gamma^*_{r+1}\nabla^r\tilde{f}_{i,j+1}\right)$$

The differences $\nabla^r f_{i,j+1}$ are then computed and replace the differences $\nabla^r\tilde{f}_{i,j+1}$ and the sums $''F_{i,j+1}$ and $'F_{i,j+1}$ are computed based on the quantity $f_{i,j+1}$.

---

† Numerical values of the coefficients in the difference and summation functions are given in Table 5-VIII.

Table 5-VIII. Numerical Values of Coefficients

Constant Subscripts	C	D	σ	σ^*	γ	γ^*
0	$\frac{1}{12}$	$-\frac{1}{2}$				
1		$-\frac{1}{12}$			$\frac{1}{2}$	$\frac{1}{2}$
2	$-\frac{1}{240}$	$\frac{1}{24}$	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{5}{12}$	$-\frac{1}{12}$
3		$\frac{11}{720}$	$\frac{1}{12}$	0	$\frac{3}{8}$	$-\frac{1}{24}$
4	$\frac{31}{60,480}$	$-\frac{11}{1,440}$	$\frac{19}{240}$	$-\frac{1}{240}$	$\frac{251}{720}$	$-\frac{19}{720}$
5		$-\frac{191}{60,480}$	$\frac{3}{40}$	$-\frac{1}{240}$	$\frac{95}{288}$	$-\frac{3}{160}$
6	$-\frac{289}{3,628,800}$	$\frac{191}{120,960}$	$\frac{863}{12,096}$	$-\frac{221}{60,480}$	$\frac{19,087}{60,480}$	$-\frac{863}{60,480}$
7		$\frac{2,497}{3,628,800}$	$\frac{275}{4,032}$	$-\frac{19}{6,048}$	$\frac{5,257}{17,280}$	$-\frac{275}{24,192}$
8	$\frac{317}{22,809,600}$	$-\frac{2,497}{7,257,600}$	$\frac{33,953}{518,400}$	$-\frac{9,829}{3,628,800}$	$\frac{1,070,017}{3,628,800}$	$-\frac{33,953}{3,628,800}$
9			$\frac{8,183}{129,600}$	$-\frac{407}{172,800}$	$\frac{25,713}{89,600}$	$-\frac{8,183}{1,036,800}$
10			$\frac{3,250,433}{53,222,400}$	$-\frac{330,157}{159,667,200}$		

Thus, starting with j = 8, that is, off the end of the known difference table, the computation proceeds cyclicly.

5.7.3 Step-size Control

At the beginning of an integration a nominal step-size Δt is supplied. At each integration step thereafter the step-size is tested to determine if its current value is adequate in order that accuracy will be maintained. Thus, after an integration step has been completed the quantity

$$V = \max_{1 \leq i \leq 3} \left| \frac{\nabla^7 f_{i,\,j+1}}{\max(y_{i,\,j+1},\, y_{min})} \right|$$

is computed. If $V \geq 10^{3-S}/(\Delta t)^2$ then the integration is continued with step-size $\Delta t/2$. Nominally, S = 12. (Analyst may change the error control tolerance by inputting 10^{3-S} with a 99-card Item 482 with preliminary data.) The number y_{min} is an input constant which is different from zero and which scales down the number V when one of the functions y_i is too near zero. Nominally $y_{min} = 0.1$. (Analyst may change y_{min} with a 99-card Item 481 input with preliminary data.)

In the event that $10^{-1-S} < V < 10^{3-S}$, the integration proceeds at the current step-size. If $V \leq 10^{-1-S}/(\Delta t)^2$ the number

$$W = \max_{1 \leq i \leq 3} \left| \frac{\nabla^8 f_{i,\,j+1}}{\max(y_{i,\,j+1},\, y_{min})} \right|$$

is computed. If $W \leq 10^{-1-S}/(\Delta t)^2$ then the routine proceeds with step-size $2\Delta t$. The constant S is chosen at the outset of the integration and is fixed thereafter. Attention is called to the fact that only the first three equations are subject to the test described above.

The mechanism for proceeding at halved or doubled step-size consists merely in retabulating the current line of differences. Thus, if it is desired to halve the step-size and $\boldsymbol{\nabla}$ represents a halved difference, we compute

$$\nabla^8 f_{i,j+1} = \frac{1}{256}\nabla^8 f_{i,j+1}$$

$$\nabla^7 f_{i,j+1} = \frac{1}{128}\left(\nabla^7 f_{i,j+1} + 448\nabla^8 f_{i,j+1}\right)$$

$$\nabla^6 f_{i,j+1} = \frac{1}{64}\left[\nabla^6 f_{i,j+1} + \left(192\nabla^7 f_{i,j+1} - 240\nabla^8 f_{i,j+1}\right)\right]$$

$$\nabla^5 f_{i,j+1} = \frac{1}{32}\left[\nabla^5 f_{i,j+1} + \left(80\nabla^6 f_{i,j+1} - 80\nabla^7 f_{i,j+1} + 40\nabla^8 f_{i,j+1}\right)\right]$$

$$\nabla^4 f_{i,j+1} = \frac{1}{16}\left[\nabla^4 f_{i,j+1} + \left(32\nabla^5 f_{i,j+1} - 24\nabla^6 f_{i,j+1} + 8\nabla^7 f_{i,j+1} - \nabla^8 f_{i,j+1}\right)\right]$$

$$\nabla^3 f_{i,j+1} = \frac{1}{8}\left[\nabla^3 f_{i,j+1} + \left(12\nabla^4 f_{i,j+1} - 6\nabla^5 f_{i,j+1} + \nabla^6 f_{i,j+1}\right)\right]$$

$$\nabla^2 f_{i,j+1} = \frac{1}{4}\left[\nabla^2 f_{i,j+1} + \left(4\nabla^3 f_{i,j+1} - \nabla^4 f_{i,j+1}\right)\right]$$

$$\nabla f_{i,j+1} = \frac{1}{2}\left(\nabla f_{i,j+1} + \nabla^2 f_{i,j+1}\right) .$$

Let $\nabla^{-1} f_{i,j+1} = {}'F_{i,j+1}$ and $\nabla^{-2} f_{i,j+1} = {}''F_{i,j+1}$

then

$$\nabla^{-1} f_{i,j+1} = \left(2\nabla^{-1} - \sum_{r=0}^{8} \frac{1}{2^{r+1}}\nabla^r\right) f_{i,j+1}$$

$$\nabla^{-2} f_{i,j+1} = \left[4\nabla^{-2} - \nabla^{-1} - \sum_{r=0}^{8} \frac{r+3}{2^{r+2}}\nabla^r\right] f_{i,j+1}$$

On the other hand if it is desired to double the step-size and ∇ represents a doubled difference the formulas are

$$\nabla f_{i,j+1} = 2\nabla f_{i,j+1} - \nabla^2 f_{i,j+1}$$

$$\nabla^2 f_{i,j+1} = 4\nabla^2 f_{i,j+1} - \left(4\nabla^3 f_{i,j+1} - \nabla^4 f_{i,j+1}\right)$$

$$\nabla^3 f_{i,j+1} = 8\nabla^3 f_{i,j+1} - \left(12\nabla^4 f_{i,j+1} - 6\nabla^5 f_{i,j+1} + \nabla^6 f_{i,j+1}\right)$$

$$\nabla^4 f_{i,j+1} = 16\nabla^4 f_{i,j+1} - \left(32\nabla^5 f_{i,j+1} - 24\nabla^6 f_{i,j+1} + 8\nabla^7 f_{i,j+1} - \nabla^8 f_{i,j+1}\right)$$

$$\nabla^5 f_{i,j+1} = 32\nabla^5 f_{i,j+1} - \left(80\nabla^6 f_{i,j+1} - 80\nabla^7 f_{i,j+1} + 40\nabla^8 f_{i,j+1}\right)$$

$$\nabla^6 f_{i,j+1} = 64\nabla^6 f_{i,j+1} - \left(192\nabla^7 f_{i,j+1} - 240\nabla^8 f_{i,j+1}\right)$$

$$\nabla^7 f_{i,j+1} = 128\nabla^7 f_{i,j+1} - 448\nabla^8 f_{i,j+1}$$

$$\nabla^8 f_{i,j+1} = 256\nabla^8 f_{i,j+1}$$

$$\nabla^{-1} f_{i,j+1} = \left(\frac{1}{2}\nabla^{-1} + \sum_{r=0}^{8} \frac{1}{2^{r+2}}\nabla^r\right) f_{i,j+1}$$

$$\nabla^{-2} f_{i,j+1} = \frac{1}{4}\left(\nabla^{-2} + \nabla^{-1} + \sum_{r=0}^{8} \frac{r+3}{2^{r+2}}\nabla^r\right) f_{i,j+1}$$

5.7.4 Integration to Specific Times

If values of the functions y_i and y_i' are required at a time different from $t_0 + j\Delta t_j$, $j = 0, 1, \cdots$, the routine is interrupted at the point $t_0 + j\Delta t_j$ just prior to the required point and one Runge-Kutta integration is performed at a step-size $h = t - (t_0 + j\Delta t_j)$. The procedure then returns to the time point $t_0 + j\Delta t_j$ and proceeds as if it had not been interrupted.

5.8 EARTH'S GRAVITATIONAL POTENTIAL MODEL

ESPOD provides the capability to calculate and sum the spherical harmonics of the Earth's gravitational potential field. The precise structure of this field depends upon the coefficients $J_{n,m}$ and phase angles $\lambda_{n,m}$ provided for the individual terms of the expansion. These coefficients are:

J_n	=	Zonal Harmonic Coefficients of order n
$J_{n,m}$, $n = m$	=	Sectorial Harmonic Coefficients
$J_{n,m}$, $m < n$	=	Tesseral Harmonic Coefficients of order n, degree m
$\lambda_{n,m}$, $m \leq n$	=	Reference phase angle for Sectorial component of Sectorial and Tesseral Harmonics.

J_n may be arbitrarily specified for $2 \leq n \leq 12$; $J_{n,m}$ and $\lambda_{n,m}$ may be arbitrarily specified for $2 \leq n \leq 6$, $1 \leq m \leq 6$. The Sectorial and Tesseral coefficients are internally converted to the form

$$S_{n,m} = J_{n,m} \sin m\lambda_{n,m}$$

$$C_{n,m} = J_{n,m} \cos m\lambda_{n,m}$$

5.8.1 Models Available in ESPOD

5.8.1.1 Model 1

ESPOD, without preliminary data input to modify the Earth potential model, automatically uses only the zonal harmonics through J_4. The coefficients stored with the program may be printed on option and verified as agreeing with accepted values. The intended coefficients in the set through J_4 are shown as Model 1 in Table 5-IX. These coefficients hold the error to within approximately 0.5 km when integrated over a 24-hour period at a 185 km altitude. The major contribution to error is due to neglecting $J_{2,2}$, resulting in a sinusoidal error with a period of 12 hours and amplitude of approximately 0.5 km.

5.8.1.2 Model 2

If alternate, more exact, or more complicated models are required, it is necessary to change the potential model with preliminary data cards. For example, if it is desired to use the zonal harmonics through J_9, that

Table 5-IX. Earth Potential Model Options

n	m	Stored in Program		Model 1		Model 2		Model 3		Model 4	
		$J_{n,m} \times 10^{+6}$	λ	$J_{n,m} \times 10^{6}$	λ	$J_{n,m} \times 10^{6}$	λ	$J_{n,m} \times 10^{6}$	λ	$J_{n,m} \times 10^{6}$	λ
2		1082.30		1082.30		1082.48*					
3		-2.3		-2.3		-2.562*					
4		-1.8		-1.8		-1.84*					
5		-0.064				-0.064					
6		0.390				0.390		Use Model 2		Use Model 2	
7		-0.470				-0.470					
8		-0.020				-0.020					
9		0.117				0.117					
2	2	2.32	-37.5					2.32	-37.5	2.32	-37.5
3	1	3.95	22.	OMIT						3.95	22.
3	2	0.41	31.							0.41	31.
3	3	1.91	51.3			OMIT				1.91	51.3
4	1	2.64	163.5							2.64	163.5
4	2	1.67	54.							1.67	54.
4	3	0.46	-13.							0.46	-13.
4	4	0.56	50.3							0.56	50.3

is, Model 2, the ZØNAL card properly punched, must be input with the preliminary data. The coefficients for J_2, J_3, and J_4 will have to be changed with 99-card input to be converted from the set using through J_4 into the set using through J_9. The coefficients from J_5 through J_9 stored in the program are intended to be proper for the set through J_9. The intended coefficients in the set through J_9 are indicated as Model 2 in Table 5-IX. As above, the set of coefficients through J_9 holds the error within approximately 0.5 km integrated over a 24-hour period at a 185 km altitude. Again, the major contribution to error is due to neglecting $J_{2,2}$ resulting in a sinusoidal error with a 12 hour period and amplitude of approximately 0.5 km.

5.8.1.3 Model 3

Either of the preceding sets of coefficients can be improved by including with them $J_{2,2}$, $\lambda_{2,2}$; the set through J_9 so modified is indicated by Model 3. This term contributes a periodic effect only and its inclusion will not disturb the internal consistency of either Model 1 or Model 2. The sectorial harmonics are called into use by inputting a SECTR card with the preliminary data.

5.8.1.4 Model 4

For further accuracy, additional harmonics may be called into use with the SECTR and TESSR cards as desired by the analyst. The fullest set of coefficients provided with ESPOD are indicated as Model 4 in Table 5-IX. Using Model 4 which includes Sectorial and Tesseral Harmonics through $J_{4,4}$ in conjunction with the Zonal Harmonics through J_9 provides accuracy of approximately 0.2 km after 24 hours of integration at an altitude of 185 km. The residual error is due principally to the uncertainties in the Sectorial and Tesseral Harmonic coefficients. Note that currently published values of the coefficients of these harmonics are tentative. If and when more accurate sets of coefficients are proposed, they may be conveniently inserted with 99-card input for test and trial purposes.

5.9 SUN-MOON GRAVITY POTENTIALS (PLANETARY GRAVITY POTENTIALS)

In its normal mode of operation ESPOD includes the perturbations due to the gravitational potentials of the sun and moon. For this purpose

it requires the ephemerides of the sun and moon as provided by the JPL planetary ephemeris tape. This tape is provided on tape LØG8.

The analyst may selectively eliminate the perturbations due to the sun and moon. If both are eliminated, the ephemeris tape is not required. The moon perturbation is eliminated with 99-card item 153 set to 0; the sun perturbation is eliminated with 99-card item 154 set to 0.

ESPOD does not include the gravitational perturbations due to other planets of the solar system.

5.10 ATMOSPHERIC DRAG

Atmospheric drag is derived as a force tangent to the direction of travel of the spacecraft, jointly proportional to a drag parameter and the density of the atmosphere.

5.10.1 Drag Parameter Model

The drag parameter may assume, on option, any of three forms:

1. $C_D A/2m$, the drag parameter
2. $C_D A/2m + K\left(\frac{t - t_e}{1440}\right)$, the drag parameter plus a secular variation proportional to time from epoch. K is the change in $C_D A/2m$ in 24 hours.
3. $C_D A/2m + K\left(\frac{1}{2}\cos^5\frac{\psi'}{2} - \frac{1}{4}\right)$, the drag parameter plus a periodic variation as a function of the earth center angle between the spacecraft and the atmospheric bulge. K is amplitude of the variation from the bulge to the back side of the earth.

5.10.2 Atmosphere Model

The atmosphere model may assume, on option, any of the four forms described in this section.

5.10.2.1 The ARDC Model Atmosphere, 1959

This model is altitude dependent only from 0 km to 700 km. It is an idealized model appropriate to the average effects during a period of uniform high solar activity. The years 1966 through 1969 are expected, on the average, to be compatible with the ARDC 59 model.

5.10.2.2 ARDC 59/Paetzold 62 Dynamic Model

This model provides for density to be time and position dependent for altitudes above 130 km. At different altitudes, the density is modeled as follows:

$0 < h \leq 130$ km	=	ARDC 1959
$130 < h \leq 150$ km	=	Faired interpolation between ARDC 1959 and Paetzold
$150 < h \leq 1600$ km	=	Paetzold Model
$1600 < h$	=	Density defined as zero

It is dependent upon solar time at the subspacecraft point, season of the year, decimetric (10.7 cm) solar flux (F_{10}) and daily geomagnetic planetary amplitude (A_p). It is uniformly applicable between the latitudes of positive and negative 60 degrees. Because of longitude dependence, in the immediate neighborhood of the poles, the density can vary rapidly, but will on the average take on an appropriate value. Because of this effect, the Paetzold atmosphere is not recommended for spacecraft with very high inclinations having their perigees near the poles, especially during magnetic storms.

For spacecraft having perigees at approximately 250 km altitude at geomagnetic latitudes less than 50 degrees, it is estimated that the density calculated from the Paetzold Model atmosphere is accurate to within ±10 to 20 percent.

5.10.2.3 U.S. Standard Atmosphere, 1962 (COESA 62 Static)

This model is altitude dependent only from -5 km to 700 km. "It is an idealized, middle-latitude, year round mean over the range of solar activity between sunspot minima and maxima .....however...(molecular weight profiles) are based upon experimentally determined values, modified slightly...." (From Part II of introduction to U.S. Standard Atmosphere, 1962).

5.10.2.4 U.S. Standard Atmosphere, 1962, Including Corrections for Top-Atmospheric Temperatures (COESA Dynamic)

This model provides for density to be time and position dependent for altitudes above 90 km. At different altitudes, the density is modeled as follows:

$$-5 < h \leq 90 \text{ km} = \text{COESA 62 Static}$$
$$90 < h \leq 700 \text{ km} = \text{COESA 62 Dynamic}$$
$$700 < h = \text{Density defined as zero}$$

This model utilizes a correction which is an empirical function of the atmospheric temperatures and the altitude. The temperature is in turn a function of the earth center angle between the spacecraft and the atmospheric bulge, decimetric solar flux (F_{10}), and daily geomagnetic planetary amplitude (A_p).

5.10.3 Application of Drag Models

The analyst has the option to decide which, among the drag parameters and atmospheres at his disposal, is best applicable to the case. Where drag is a small factor, because of the shortness of the interval of tracking or the very high altitude of perigee, the unvaried drag parameter will suffice.

When long intervals of tracking occur in a rapidly decaying situation, secular variation in drag can be observed and may be provided for. Secular variation may be used with all of the given atmosphere models.

When moderate intervals of tracking occur with a high eccentricity and with perigee near the atmospheric bulge, the variation in density due to position in the bulge can be larger than the variation due to altitude. This positional variation can be applied to either of the static atmospheres. It provides a quasidynamic effect without requiring the special inputs of F_{10} and A_p (Table 5-X).

Table 5-X. Applicability for Combined Atmospheres and Drag Models

	ARDC 59	COESA Static	Paetzold	COESA Dynamic
$C_D A/2m$	Applicable	Applicable	Applicable	Applicable
$C_D A/2m + K(t - t_o)$ 1440	Applicable	Applicable	Applicable	Applicable
$C_D A/2m + Kf(\psi')$	Applicable	Applicable	Not Applicable	Not Applicable

5.10.4 Source of A_p and F_{10}

The Paetzold model and the COESA dynamic model require input of the parameters F_{10} and A_p. Average values may be entered with a single APF10 card identified by the day of epoch. With average values of A_p and F_{10}, a specially tailored static atmosphere with position dependency is obtained. Daily values may also be entered with multiple APF10 cards identified by day of current year; one card is required for each four days of integration, during both differential correction and position prediction.

The most convenient source of current values of A_p and F_{10} is the North Atlantic Radio Warning Service, Fort Belvoir, Virginia. The most convenient source of one to three-day predictions of A_p and F_{10} is the English language TWX messages sent out from the Air Weather Service Scientific Services Office, Scott Air Force Base, Belleville, Illinois. (A_{Fr}, the Fredericksburg geomagnetic amplitude, is used as a real-time approximation to A_p). These messages have the format shown in Figure 5-5. "SOLAK" means A_k (actually A_{Fr}), and "SOLRF" means F_{10}. A_k is expressed in the units 2γ (or 2×10^{-5} gauss). F_{10} is given in watts per square meter per cycle per second. The actual measurements of F_{10} are made by the National Research Council, Ottawa, Canada. The measurements of A_{Fr} are made at the Fredericksburg Geomagnetic Observatory, Corbin, Virginia. The values of A_p are computed at the University of Göttingen, West Germany, and these values become available several months after the date to which they apply.

5.10.5 Input Control

In order to incorporate atmospheric drag in the solution it is mandatory that the analyst coordinate inputs on the following preliminary data cards:

DRAG

Specify an initial value of $C_DA/2m$. If drag variation is required.

Specify secular or periodic drag variation

Specify an initial value of K

Specify atmosphere model

MEASUREMENT MESSAGE

```
00  150110Z
FM AGIWARN BUSTANS FT BELVOIR VA
TO RUWHSJ/6594 AEROSPACE TEST WG SUNNYVALE CALIF
BT
UNCLAS WASHAWS 05237
ACTION:  ACES
DAY                 SOLAK                  SOLRF
14                   006                    ,075
BT
15/0137Z
```

PREDICTION MESSAGE

```
00  150100Z
FM AWS SCOTT AFB ILL
TO RUWHSJ/6594 AEROSPACE TEST WG SUNNYVALE CALIF
BT
UNCLAS AWSSS 15-M-14
ACTION:  ACES
DAY                 SOLAK                  SOLRF
15                   007                    077
16                   010                    082
17                   012                    091
BT
15/0130Z
```

Figure 5-5. Example of Measurement and Prediction Messages for A_k (SOLAK) and F_{10} (SOLRF)

CAT1

Specify solution for $C_DA/2m$

If required, specify solution for K

BNDS (if nominal bounds are inappropriate)

Secify initial bound on differential correction in $C_DA/2m$. If required, specify initial bound on differential correction in K.

APF10 (if dynamic atmosphere is used)

Specify day of year, Jan 1 = 1

Specify A_p

Specify F_{10}

(Provide one card for each four days of integration, unless average value is used)

5.10.6 Interpretation of Drag Parameter

ESPOD accepts the density provided by the model atmosphere as a function of the appropriate inputs. The drag parameter plus its variation is applied as a multiplier to this density. ESPOD will solve for those values of the drag parameter and variation which minimize the root mean squares of the residuals. The resulting drag parameter is a characteristic of the spacecraft only if the atmosphere model is accurate. The effect of including atmospheric drag which is solved from empiric data is summarized as follows:

a) Atmosphere Model is perfectly accurate.

1. Solved $C_DA/2m$ is average characteristic of spacecraft

2. Drag is accurately represented, if the spacecraft has a constant $C_DA/2m$.

b) Atmosphere Model density is a constant multiple of real atmospheric density in region of interest.

1. Solved $C_DA/2m$ is a product of this multiple and the actual average $C_DA/2m$ of spacecraft.

2. Drag is accurately represented if the spacecraft has a constant $C_DA/2m$.

c) Atmospheric Model is anomalous with respect to real atmosphere.

1. $C_D A/2m$ is a corrective multiple which minimizes residuals

2. Drag is represented inaccurately to a degree depending on the anomaly of the model atmosphere.

5.11 RADIATION PRESSURE MODEL

ESPOD provides a radiation pressure model to estimate the contribution of solar radiation pressure to the spacecraft acceleration. The model as it stands is appropriate to spacecraft which are not subject to extensive eclipsing, that is it is always applicable to high altitude earth satellites, lunar probes, and interplanetary spacecraft. If desired, the model may be used quite accurately for earth orbiting spacecraft which are eclipsed for larger fractions of their orbits by adjusting the area to mass ratio.

To call the radiation pressure model into operation, the analyst must submit a RADPR card on which he has entered an estimate of the spacecraft's area and mass. The size of the contribution of radiation pressure to the acceleration relative to the other contributions indicates that it is not a very significant effect for short intervals of tracking. Approximate values of area and mass will typically result in an adequate estimated contribution to the acceleration.

If the area and mass are known accurately, it is preferable to use them. If they are to be adjusted for eclipsing, the area to mass ratio is multiplied by (1-f) where f is the fraction of time the satellite is in eclipse. The model uses a solar radiation flux of 1369 watts per square meter (at 1 A.U. from the sun).

The input quantities which determine the radiation pressure effects are the mass and area of the spacecraft. However, the "effective area," A_{eff}, and not the area of the spacecraft is the input quantity on the RADPR card. The effective area is defined:

$$A_{eff} = a(1 + \gamma B)$$

where

A = Area of the spacecraft

γ = Percentage of radiation reflected from the surface according to some reflection law, B.

$$B = \begin{cases} 1 \text{ for specular reflection} \\ 2/3 \text{ for diffuse reflection} \end{cases}$$

The quantity γB is usually determined in the laboratory. The minimum value of $A_{eff} = A$; and the maximum value, 2A. The quantity $(1 + \gamma B)$ almost never exceeds 1.20.

5.12 DIFFERENTIAL *a priori* S MATRIX CORRECTION CONTINUATION

ESPOD provides a special option which permits the analyst to input an *a priori* $A^T A$ matrix, or for brevity, S matrix (SMAT cards). The option is applicable when observational data is processed in batches and more than one batch is required for a single differential correction.

Figures 5-6 and 5-7 respectively show how the first and second batches are processed. Certain restrictions implied in the method are:

- The initial conditions must remain the same
- The solution vectors must remain the same
- The set of weights applied to observations 2 must be the same as that applied to observations 1.

Any further batches are processed as in Figure 5-7 with the previously obtained $S_{1,\ 2}, \ldots$ matrix input as an *a priori* condition.

Because of the restriction that initial conditions remain the same it is not meaningful to let the program iterate on the new batch of observations with a fixed *a priori* $A^T A$ matrix. The differential correction performed as a function of O_1 plus O_2 (see Figure 5-6) results in a new set of initial conditions IC2. These new initial conditions are not appropriate to the *a priori* S_1 matrix and thus successive iterations do not fulfill restriction 1. If it is required to iterate on O_1 plus O_2, it is necessary to include both O_1 and O_2 as observations in the input data.

S-Matrices may be obtained in a form compatible with ESPØD input from two sources.

1) On option, every differential correction iteration will punch out the current S-matrix relative to the best elements to date and will also punch those elements (JDC column 43 is set to 1; $\alpha\delta\beta ARv$ elements and ITIME cards are correctly provided if JDC column 46 is set to 0).

2) On option, the covariance matrix is updated by ESPØDEPH to a specified time (JDC column 55 is set to 1), and the inverse covariance matrix, which is the S-matrix, is calculated and punched (JDC column 56 is set to 1). This S-matrix is relative to the osculating elements at the time of update. These must be transcribed from the ephemeris printout to ICØND and ITIME cards for input as applicable initial conditions.

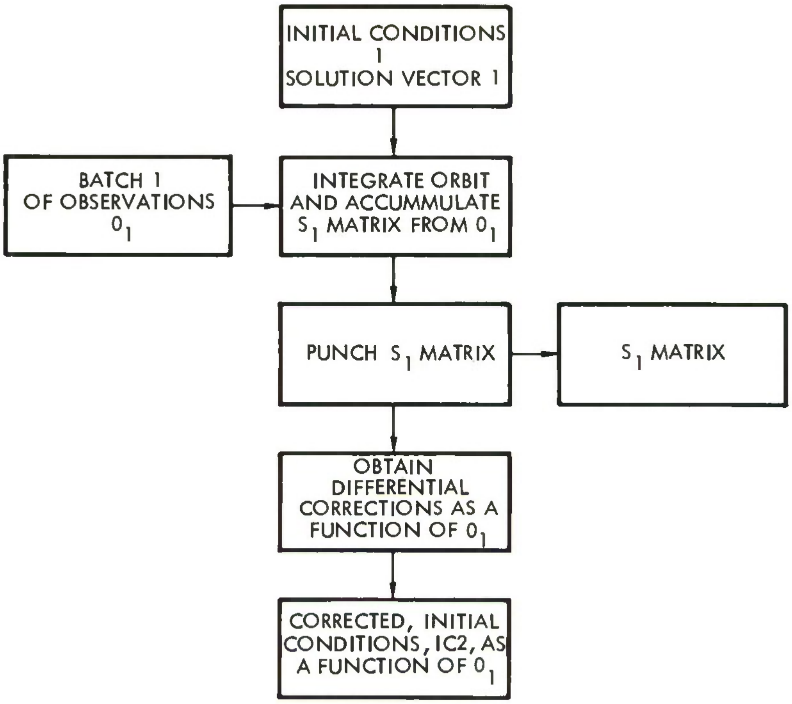


Figure 5-6. Processing Observations Without a priori S, A^TA Matrix

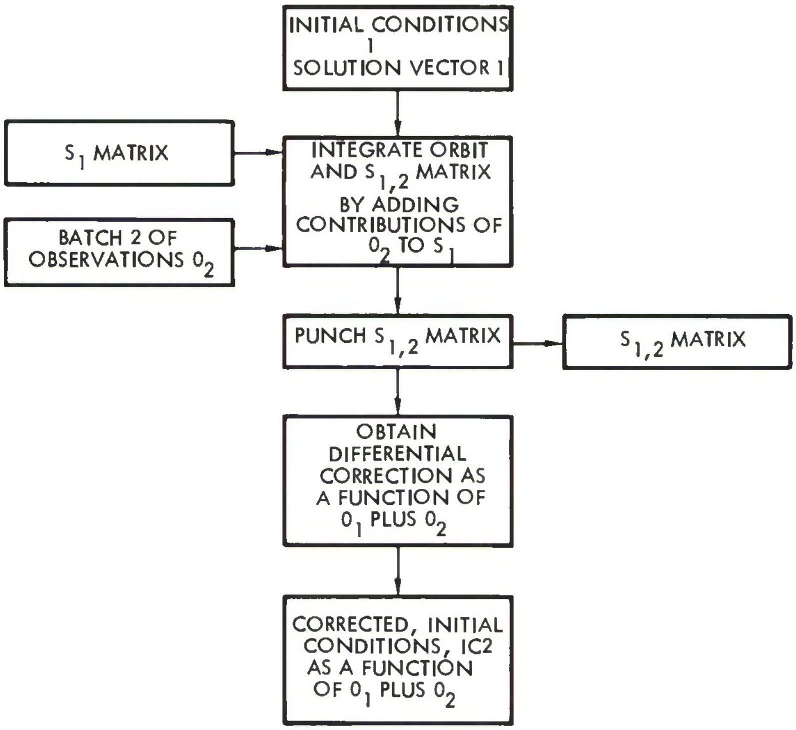


Figure 5-7. Processing Observations to Accumulate Further An A^TA Matrix

5.13 ESPØDEPH CONTROL

The spacecraft ephemeris is generated in ESPØDEPH. Thus, ESPØDEPH requires all the preliminary initial conditions and callouts for perturbations to the force model. The preliminary data and control data items appropriate to ESPØDEPH are listed in Table 5-XI. Note that items IA, IH, and IIA are mandatory.

5.13.1 Source of Preliminary Data

ESPØDEPH may be used under any of three conditions:

1) Automatically following ESPØDDC
2) On a conditioned or conditional start
3) On a "cold-start" without ESPØDDC

For each of these conditions, the preliminary or control data, or both, may come from different sources. The sources are indicated in Figure 5-8, depending on the path through the flow chart. If a previous ESPØDDC run is assumed, which either automatically preceeds ESPØDEPH or which has left its output behind for a subsequent conditioned start, the preliminary data is taken from Tape LØG7. In the automatic case, there is no opportunity to insert new preliminary data on cards. In the conditioned start case, it is mandatory to specify the ephemeris time points and any particulars of the acceleration perturbations may be changed for special purposes. For a cold start, ESPØDEPH run only, all mandatory preliminary data specifying initial conditions, ephemeris time points must be specified, and any further optional data may be specified. If covariance matrix update is required, it is mandatory to input an _a priori_ convariance matrix with UPMAT cards.

5.13.2 ESPØDEPH Output

ESPØDEPH prints out at each ephemeris point the position and velocity of the spacecraft in element sets having the following forms and coordinates:

- Orbit plane classical elements
- Geocentric polar spherical
- Geocentric Cartesian
- Indeterminacy free
- Selenocentric and heliocentric Cartesian

Other miscellaneous data is also given for convenience:

- Callendar date and time
- Minutes from epoch
- Julian date
- Geodectic latitude, longitude, altitude
- Time to next nodal crossing
- Argument of latitude
- Altitude at apogee
- Altitude at perigee
- Period

If the covariance matrix is updated, the standard deviations and correlations are given for elements in the following forms and coordinates:

- Orbit plane classical elements
- Geocentric polar spherical
- Cartesian (Earth centered inertial)

If the covariance matrix is updated, the standard deviations and correlations for position are given in the following coordinates:

- UVW
- Axes of the position error ellipsoid

5.13.3 Specification of Time Points

Time points for ephemeris prediction may be input in any of three mutually exclusive modes:

1) Special format cards (DAC cards) giving the time point for prediction and a test Cartesian position, which has been computed elsewhere, of the spacecraft at that time.

   NOTE: ESPØDEPH calculates Cartesian position with X axis directed toward the true vernal equinox at $0^h.0$ on the day of epoch given with the initial conditions. This may be different from the coordinate system in which the previously computed test value is given.

2) Ephemeris table generation control (DELTT cards) which enable the calculation and printing of the ephemeris at equally spaced intervals.

3) Prediction at specified times is enabled with the PRDCT cards.

Table 5-XI. ESPØDEPH Preliminary and Control Data Options

I. Preliminary Data

A. Initial Orbit Elements and Epoch (Mandatory)

B. Atmospheric Drag Model

1) Drag parameter $C_D A/2m$

a) Drag variation coefficient K
b) Drag variation model

2) Model atmosphere

a) APF10 data

C. Geopotential Model Perturbations

D. Solar-Lunar Perturbations

E. Radiation Pressure Model

F. Integration Control

G. Covariance Matrix

1) Results from differential correction
2) A priori covariance matrix

H. Time Points for Ephemeris (Mandatory)

1) DAC cards
2) ESPØD preliminary data cards

II. Control Data

A. Select ESPØDEPH (JDC Column 51) (Mandatory)

B. Select or Omit ESPØDDC (JDC Column 41)

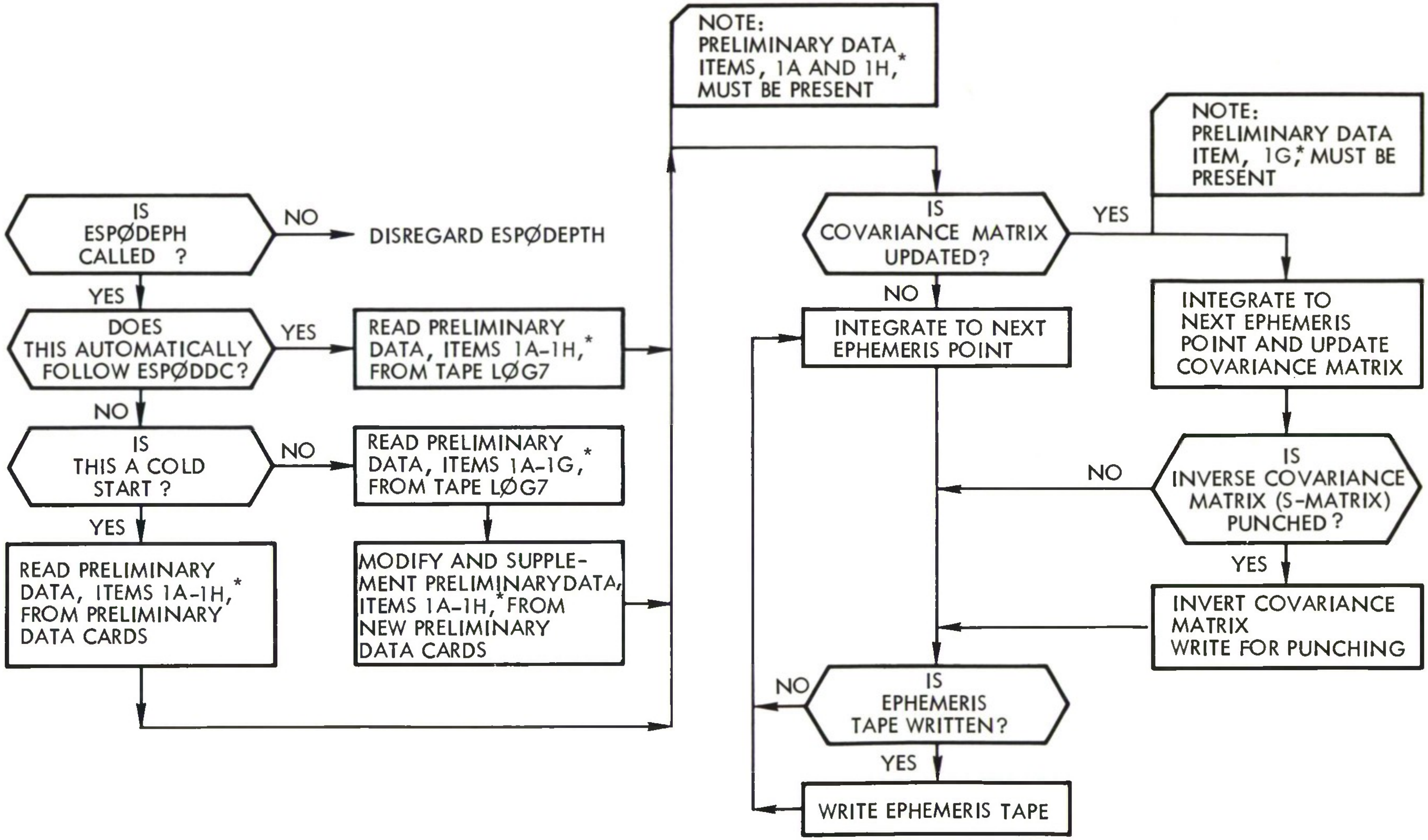
C. Select "Cold Start" or Tape LØG7 Preliminary Data (JDC Column 30)

D. Select or Omit Covariance Matrix Update (JDC Column 55)

E. Select DAC Cards for Time Point Specification (JDC Column 52)

F. Generate or Disregard Ephemeris Tape (JDC Column 53)

G. Punch or Disregard Inverse Covariance Matrix Update (JDC Column 56)



*ITEMS FROM TABLE 5-XI

Figure 5-8. ESPØDEPH Control Logic

5.14 CONDITIONED START

The conditioned start options (JDC column 30 = 1 or 2) permit the preliminary data for a run to be taken from the tape record left behind by the last iteration of some preceding differential correction. The tape left on TAPE LOG 7 after a differential correction contains the data necessary for this type of continuation.

1) Restarting an iteration after interrupt or machine failure, without necessitating the reading in of all preliminary data, observations, and sensors again and without repeating already performed iterations.

2) Extending the immediately previous differential correction with more iterations, possibly with the solution vector or force model modified.

3) Performing a prediction with the results of immediately previous differential correction.

5.14.1 Restrictions

As mentioned in Item 2) above, it is permissible to modify some of the preliminary data associated with the previous differential correction with new preliminary data cards input with the JDC card which calls for the conditioned start. The following restrictions apply:

- The object must remain the same.
- The initial conditions may be changed, but this nullifies any benefit from a previous differential correction run.
- The solution vector may be changed, but if drag is added to the solution, and a drag model was not previously provided, it must be provided with the new preliminary data.
- The force model may be changed, but if this is an ESPØDEPH run only, the new circumstances will not be appropriate to the elements derived by the differential correction.
- New observations may not be added.
- New sensors may not be added. However, sensor type data (STYPE cards) and standard deviations (SIGMA cards) may be modified.

5.14.2 Conditional Start

Conditional start is a suboption of conditioned start. The conditioned start option is provided principally to guard against doing a prediction using initial conditions which, because of the circumstances of their formation, may be worse than the best available. In the conditional start mode, the run will not proceed if two conditions were present when the source differential correction terminated:

1) Convergence was not obtained on the final iteration.

2) The next to last iteration was a diverging step. This situation is the result of termination on the NITER test in trials 2, 3, or 4 while attempting to obtain a converging step by reducing bounds (see Figure 5-4).

The points where exit to B can occur because of the "NITER complete" test, within the blocks titled "Trial 2, for E_2," "Trial 3 for E_2," and "Trial 4 for E_2," are the exit points which prohibit a subsequent conditional start.

5.15 IMPACT TESTS

ESPOD tests for impact of the spacecraft in a differential correction run and in an ephemeris generation. For an ESPØDEPH run, the program tests for impact with the earth's surface, defined as a sphere of 6378.165 km in radius. If impact occurs during an ESPØDEPH run, the process is terminated.

Impact in a differential correction run (ESPØDDC) occurs if the radius vector to the vehicle falls below 6453.165 km, i.e., below an altitude in the approximate range of 66 to 84 km. If impact occurs during an ESPØDDC run, observations which have already been processed are used to obtain some kind of correction. If the situation is not hopeless, the correction will result in a better set of initial conditions which will, on the next iteration, carry the orbit farther.

5.16 COVARIANCE MATRIX

One of the ESPOD outputs is the variance-covariance matrix, which expresses the errors in the solved parameters. This matrix can be interpreted in a <u>statistical</u> or a <u>nonstatistical</u> way. For the sake of discussion, suppose that the parameters solved for are α, β, δ, A, R, v, and consider the interpretation of the upper lefthand corner element of the covariance matrix, namely σ_α^2, the variance in α.

5.16.1 Statistical Interpretation

Assumptions:

a) The mathematical model is exact, that is, if there were no errors in the observations, the result would be the perfect answer, zero residuals, and zero root mean squares.

b) The observations are subject to uncorrelated errors with mean zero.

c) The standard deviations of the errors in the observations are known. They are equal to the standard deviations which were input to the program for use as reciprocal weights.

d) The output errors are linear functions of the input errors.

It follows from all of these assumptions that σ_α^2 is the variance in the parameter α. If it is also assumed that the input errors are normal, the output errors are also normal and regions of confidence can be constructed on the variables α, δ, β, A, R, v using multivariate normal distribution theory.

5.16.2 Nonstatistical Interpretation

From another standpoint, σ_α^2 expresses the sensitivity of the output value to the input data. The entire matrix could be output which gives the partial derivatives of the output α, δ, β, A, R, v, $C_D A/2m$, K with respect to each input observation.

Since there are usually a great many observations, this would involve more output than could be typically assimilated; the covariance matrix can be regarded as a summary of this large matrix.

5.16.2.1 Using Assumption d) Above

The following question can be asked: "How much would the output value of α change if the input data changed by a certain amount?" The value σ_α^2 is directly related to this question under the following rule: "If the root mean square of the weighted changes in the input data is less than K, then the resulting change in α will be less than $\sigma_\alpha K\sqrt{n}$, where n is the number of observations. The weights used here are the inverses of the standard deviations estimated for the sensors. To make this interpretation of the covariance matrix only assumption d) is used.

5.16.2.2 Using Assumptions a) and d) Above

By adding assumption a) one can get a stronger statement, namely: If the input data differs from the true data by errors with weighted root-mean-square less than K, then the output α differs from the true α by less than $\sigma_\alpha K\sqrt{n}$.

5.16.3 Comparison

The comparison of statistical and nonstatistical interpretations will lead to quite different estimates in the errors. In the first interpretation, σ_α is a reasonable estimate for the error in α, and in the second interpretation $\sqrt{n\sigma_\alpha}$ is the estimate. Since n may typically be 1000, the estimates are quite different.

The question of which estimate to accept is a philosophical one. Input errors may be distributed "nicely" with mean zero and hence largely cancel each other out or they may be distributed in the worst possible way. As an argument for the second interpretation, it may be shown that there is a set of worst possible changes which one could make in the data, with RMS = 1, such that the change in α is actually $\sigma_\alpha \sqrt{n}$. (In mathematical jargon "the inequality is sharp.") Also, this worst possible set of changes is not a particularly diabolical set; in some cases, the worst possible set is simply a constant bias.

6. DIFFERENTIAL CORRECTION PROCESS

The discussion has two parts. The first part describes the least squares problem and the differential correction solution in an abstract mathematical way. The second part shows the relation between the least squares problem and orbit determination.

6.1 MATHEMATICAL DESCRIPTION

6.1.1 The Problem

A set of N functions of M variables is given.

$$\begin{aligned} &f_1 (a_1, \cdots, a_M) \\ &f_2 (a_1, \cdots, a_M) \\ &f_N (a_1, \cdots, a_M) \end{aligned} \tag{1}$$

The problem is to choose the values of $a_1, a_2, \cdots a_M$ which minimize the expression

$$\left[f_1 (a_1, \cdots, a_M)\right]^2 + \left[f_2 (a_1, \cdots, a_M)\right]^2 + \cdots + \left[f_N (a_1, \cdots, a_M)\right]^2 \tag{2}$$

The solution vector is the weighted least squares solution.

The problem is more tersely stated in vector notation. Given a vector function f(a) of a vector a, find a such that $\|f(a)\|^2$ is minimized.

6.1.2 The Solution

Suppose that an approximate solution $a_1^o, a_2^o, \cdots, a_M^o$ is given. Expand each function f_k in a Taylor series about the point $a_1^o, \cdots, a_M^o$, and drop all but the linear terms to obtain an approximate expression for f_k in a neighborhood of the point $a_1^o, \cdots, a_M^o$:

$$f_k(a_1, \cdots, a_M) \cong f_k\left(a_1^o, \cdots, a_M^o\right)$$

$$+ \sum_{j=1}^{M} \frac{\partial f_k}{\partial a_j} (a_j - a_j^o), \; k = 1, \cdots, N \tag{3}$$

The partial derivatives are evaluated at the point $a_1^o, \cdots, a_M^o$.

Now consider the problem of minimizing the approximate expression for the sum of weighted squares of the functions f_k, i.e., consider the problem of finding $a_1, \cdots, a_M$ so as to minimize the expression

$$\sum_{k=1}^{N} \left[f_k\left(a_1^o, \cdots, a_M^o\right) + \sum_{j=1}^{M} \frac{\partial f_k}{\partial a_j}\left(a_j - a_j^o\right)\right]^2 \tag{4}$$

Differentiating the expression with respect to a_j and setting the result equal to zero shows that the required $(a_1, \cdots, a_M)$ satisfy the equation

$$\sum_{j=1}^{M} \left(a_j - a_j^o\right) \sum_{k=1}^{N} \frac{1}{\sigma_k^2} \frac{\partial f_k}{\partial a_i} \frac{\partial f_k}{\partial a_j} = - \sum_{k=1}^{N} \frac{1}{\sigma_k^2} \frac{\partial f_k}{\partial a_i} f_k\left(a_1^o, \cdots, a_M^o\right) \tag{5}$$

$$i = 1, \cdots, M$$

This is a system of M linear equations in M unknowns. It is customary to solve them not for the $(a_1, \cdots, a_M)$ directly but for the "differential corrections."

$$x_j = a_j - a_j^o, \; j = 1, \cdots, M \tag{6}$$

The differential correction method can now be described as follows. Start with an approximate solution $a_1^o, \cdots, a_M^o$ and compute

$$f_k\left(a_1^o, \cdots, a_M^o\right); \; k = 1, \cdots, N$$

$$\frac{\partial f_k}{\partial a_j}\left(a_1^o, \cdots, a_M^o\right); \; k = 1, \cdots, N, \; j = 1, \cdots, M \tag{7}$$

Form the coefficients

$$c_{ij} = \sum_{k=1}^{N} \frac{\partial f_k}{\partial a_i} \frac{\partial f_k}{\partial a_j}; \; i, j = 1, \cdots, M$$

$$d_i = - \sum_{k=1}^{N} \frac{\partial f_k}{\partial a_i} f_k\left(a_1^o, \cdots, a_M^o\right); \; i = 1, \cdots, M \tag{8}$$

and solve the system of linear equations

$$\sum_{j=1}^{M} c_{ij}x_j = d_i; \; i = 1, \cdots, M \tag{9}$$

for $x_1, \cdots, x_M$. Then

$$a_j^1 = a_j^o + x_j; \; j = 1, \cdots, M \tag{10}$$

gives the next approximate solution. The values $a_1^1, a_2^1, \cdots, a_M^1$ replace $a_1^o, \cdots, a_M^o$ and the process is repeated as often as necessary.

The process can be stated more tersely in matrix notation. An approximation a^o is given. Compute the vector $f^o = f(a^o)$, and compute the matrix of partial derivatives A. The k, j element of A is given by

$$a_{kj} = \frac{\partial f_k}{\partial a_j} \tag{11}$$

Calculate the matrix C and the vector d:

$$C = A^T A \tag{12}$$

$$d = -A^T f^o \tag{13}$$

Solve the linear system $Cx = d$ for x, and let $a^1 = a^o + x$ replace a^o. Repeat as often as necessary.

6.1.3 <u>Modified Differential Corrections</u>

The method just described is the method generally given in textbooks, but it is not adequate from a theoretical or practical standpoint.

Theoretically, it requires strong hypotheses to prove convergence; practically, it is found that many cases diverge. The problem is that the "differential corrections" may not be differential. More exactly, the corrections x_j may be so large and the Taylor series expansion so poor that the sum of squares increases instead of decreases on each iteration.

The following modification is made to improve the convergence of the process. Instead of choosing $x_1, x_2, \cdots, x_M$ to minimize the expression

$$F = \sum_{k=1}^{N} \left[f_k \left(a_1^o, \cdots, a_M^o \right) + \sum_{j=1}^{M} \frac{\partial f_k}{\partial a_j} x_j \right]^2 \tag{14}$$

we choose $x_1, \cdots, x_M$ to minimize F under the side condition that

$$\sum_{j=1}^{M} \left(\frac{x_j}{B_j} \right)^2 \leq 1 \tag{15}$$

where $B_1, \cdots, B_M$ is a set of positive numbers. The side condition insures that the corrections x_j will be reasonably small; in particular, it insures that

$$\left| x_j \right| \leq B_j \tag{16}$$

The numbers $B_1, \cdots, B_M$ are called "bounds." If they are small, they tend to make the program converge in a slow but sure manner; if they are large they tend to make the program select larger and riskier corrections. The program starts out with a set of bounds which may be prescribed by the analyst, and automatically increases or decreases the bounds as the iteration continues. If an iteration fails, the program decreases the bounds. If an iteration works as predicted, then the program increases the bounds.

In matrix notation, the modified differential correction selects a vector x which minimizes $\left\| f^o + Ax \right\|^2$ under the side condition that $x^T B^{-2} x \leq 1$, where B is a diagonal matrix with positive diagonal elements B_j.

6.1.4 Forming the Coefficient Matrix

The matrix

$$C = A^T A \tag{17}$$

could be computed by forming the A matrix in the computer, transposing it, and multiplying by A with a standard matrix multiply routine. However, this method would waste time and computer memory. Instead, the matrix is formed by

$$C = \sum_{k=1}^{N} r_k^T r_k \tag{18}$$

where r_k is the k^{th} row of the A matrix. Each time a row of A is formed, the matrix $r_k^T r_k$ is added to C. This means that there is no need to store the A matrix at all, and hence an indefinite number of rows (or observations) can be used.

Similar remarks hold for the vector d, which is computed by

$$d = - \sum_{k=1}^{N} r_k^T f_k^o \tag{19}$$

6.1.5 Predicted Sum of Squares

The quantity

$$F = \left\| f^o + Ax \right\|^2 \tag{20}$$

is an estimate of the quantity

$$\widetilde{F} = \left\| f\,(a^o + x) \right\|^2 \tag{21}$$

Since F can be computed before $\widetilde{F}$ in the process, it is regarded as a prediction of $\widetilde{F}$. That is, at the time of computing the correction x, one computes $\left\| f^o + Ax \right\|^2$, which is a prediction of what will happen when the correction is actually applied.

Since the matrix A is generally not available, F is computed by the equivalent form

$$F = (A^T Ax, \ x) + 2 \ (A^T f^o, \ x) + \|f^o\|^2 \tag{22}$$

$$= (Cx, \ x) - 2 \ (d, \ x) + \|f^o\|^2 \tag{23}$$

The quantity $\|f^o\|^2$ is computed at the same time C and d are computed:

$$\|f^o\|^2 = \sum_{k=1}^{N} \left(f_k^{\ o}\right)^2 \tag{24}$$

Then F can be computed from the known quantities C, d, and $\|f^o\|^2$.

6. 1. 6 <u>Solution for the Differential Correction</u>

The unmodified form of the differential correction process requires the solution of the system of linear equations.

$$A^T Ax = -A^T f_o \tag{25}$$

In general, a system of M linear equations in M unknowns may not have a solution. One might then inquire what the differential correction method does if no solution exists for the system (25). It turns out that this is not a problem, since a solution of (25) always exists. If the matrix $A^T A$ is singular, there is still a solution; in fact, there are an infinite number of solutions. The situation is summarized by the following theorem.

a) $\|(Ax + f^o)\|^2$ is minimized if and only if x satisfies

$$A^T Ax = -A^T f^o$$

b) There exists a solution x of the equation

$$A^T Ax = -A^T f^o$$

For the modified differential correction method, the situation is slightly more complicated. Define x (λ) as a solution of the equation.

$$(A^T A + \lambda B^{-2}) \ x = -A^T f^o \tag{26}$$

Consider the problem of minimizing $\left\|f^o + Ax\right\|^2$ under the side condition that $x^T B^{-2} x \leq 1$. It is possible to prove the following theorem: (a) A solution to (26) always exists. If $\lambda \neq 0$, the solution is unique; (b) If there is a solution $x = x(0)$ of (26) for $\lambda = 0$ which satisfies the side condition, then x is a solution of the minimum problem; (c) If there is no solution of (26) which satisfies the side condition, then there is a $\lambda_o > 0$ such that $x = x(\lambda_o)$ satisfies $x^T B^{-2} x = 1$. Then $x(\lambda_o)$ is a solution of the minimum problem.

6.2 APPLICATION TO ORBIT DETERMINATION

The orbit determination problem can be expressed as a problem of minimizing

$$\left\|f(a)\right\|^2 = \sum_{k=1}^{N} \left[f_k(a_1, \cdots, a_M)\right]^2 \tag{27}$$

provided that the parameters $a_1, \cdots, a_M$ and functions $f_1, \cdots, f_M$ are interpreted in the following way.

The components of the vector a are the unknowns to be solved for. In the basic problem solved by ESPOD, there are M = 6 parameters given by the initial conditions

$$a_1 = \alpha$$
$$a_2 = \delta$$
$$a_3 = \beta$$
$$a_4 = A$$
$$a_5 = R$$
$$a_6 = v$$

In other problems, the number M of parameters may be greater than 6, and the components of "a" may include drag parameters, station location errors, data biases, etc.

The functions f_k, $k = 1, \cdots, N$ are the weighted differences between the observed data and the computed data. For example, if the first observation is a range measurement R_M from a particular station at a

particular time, and if R_C is the computed value of the range for that station and time, then the first component of f would be

$$f_1 = \frac{R_C - R_M}{\sigma} \tag{28}$$

where $1/\sigma$ is the weight assigned to that particular observation. The other components of the vector f are defined similarly.

The weighted partial derivative matrix A is obtained by differentiating the computed measurements with respect to the parameters and multiplying by the appropriate weight. For example, the first row of the A matrix would be given by

$$a_{ij} = \frac{1}{\sigma} \frac{\partial R_C}{\partial a_j}, \; j = 1, \cdots, M \tag{29}$$

Note that the calculation of the A matrix does not depend on the measured value R_M. This fact permits one to make error studies, which are generally based on the matrix A, without having any actual measured data, but using estimates of standard deviations applied to them.

7. COORDINATE SYSTEMS

This section describes and illustrates the various coordinate systems used by ESPOD either in receiving or manipulating data, or in presenting the results.

The following are definitions of terms applicable to the different coordinate systems:

Vernal Equinox:	That point of intersection of the ecliptic and celestial equator where the sun crosses the equator from south to north in its apparent annual motion along the ecliptic
Equator:	The great circle intersection of the celestial sphere and a plane containing the center of mass perpendicular to the rotating axis of the earth
True of (Epoch or Date)	The actual position at a given time of the vernal equinox including both precession and nutation
Mean of (Epoch or Date)	A fictitious equinox whose position is that of the vernal equinox at a particular time with the effect of a nutation removed
Osculating Elements:	The elements of an instantaneous orbit which are tangent to the actual trajectory, having the same position and velocity at that time
Date:	An exact time; e.g., the date of an observation is the exact time at which it was made
Epoch:	Some initial reference instant of time

Radar observations, $(R, A, E, \dot{R})$ are taken and updated in the sensor dependent coordinate system in which the axis is the actual vernal equinox, i.e., true equinox of date. These observations are used by ESPOD in the form in which they are reported.

The photographic telescope (Baker-Nunn cameras) observations (α, δ) are taken in a sensor dependent coordinate system. The photographs or plates are "reduced" by means of right ascension and declination grids for that portion of sky which was photographed. The reductions fall into two categories. The first is called "field reduced" and the second "precision reduced". The field reduced observations give right ascension and

declination in a coordinate system in which the axis is the mean equinox of 1855.0 (beginning of the Julian year) for ψ^* (geodetic latitude) $> -22^{\circ}$ or 1875.0 for $\psi^* < -22^{\circ}$. The precision reduced observations (which are reduced by the Smithsonian Observatory) give right ascension in a coordinate system in which the x-axis is directed to the mean equinox of date; i.e., corrected for precession to the time of observation only.

The reduced observations are handled in various ways by ESPOD. The precision reduced data are updated by ESPØD to a coordinate system which is true of day of epoch and mean of date. Stated simply, the observations are corrected for precision and nutation to the epoch, and for precession to the date of observations. Hence, the only correction which is not included is the nutation correction from the epoch to the date of observations. This period usually does not exceed ten days.

The field reduced data can be processed by ØRCØN which processes the data to true of date, writes on tape, and feeds the data to ESPØDDC. If the field reduced data are punched on cards mean of 1855 or 1875, the cards are processed by ESPØD to give the observations in true equinox of 00h day of epoch before being used by ESPØDDC.

Figure 7-1 summarizes the above discussion.

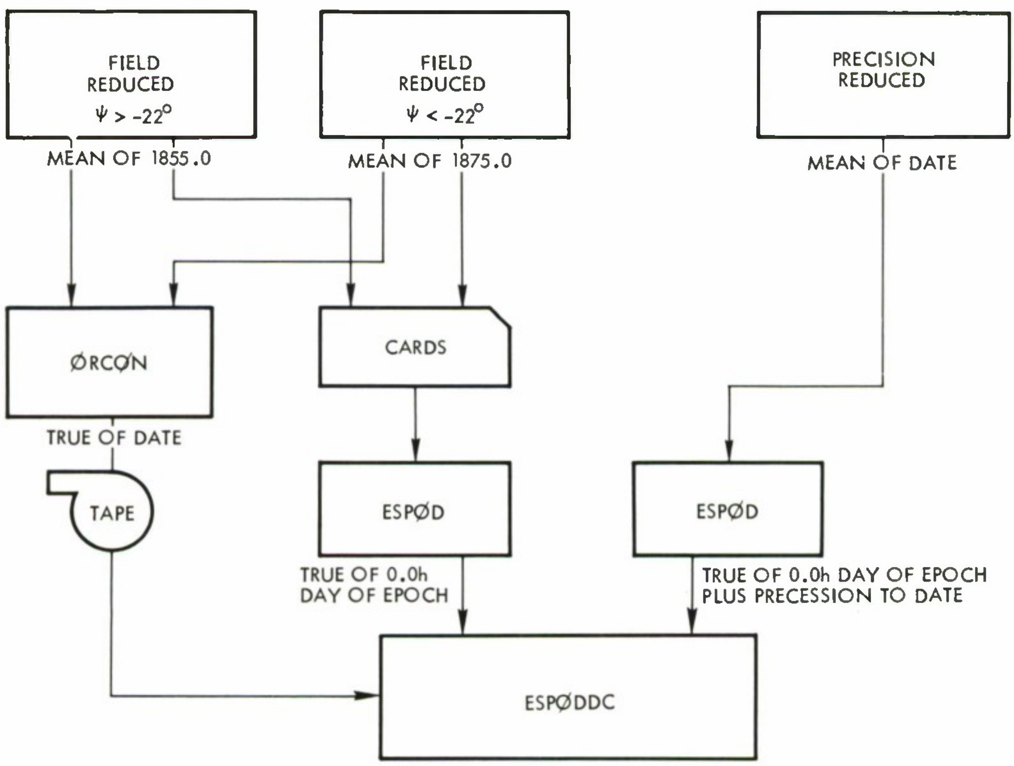
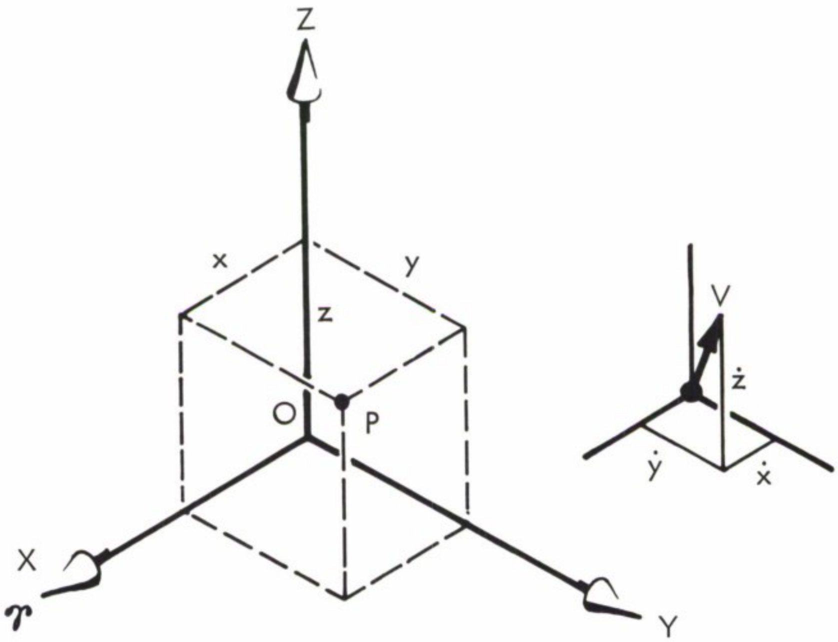


Figure 7-1. Processing Baker-Nunn Data

7.1 EARTH CENTERED INERTIAL CARTESIAN SYSTEM

The position and velocity of a body at point P are $P = P(x, y, z, \dot{x}, \dot{y}, \dot{z})$.



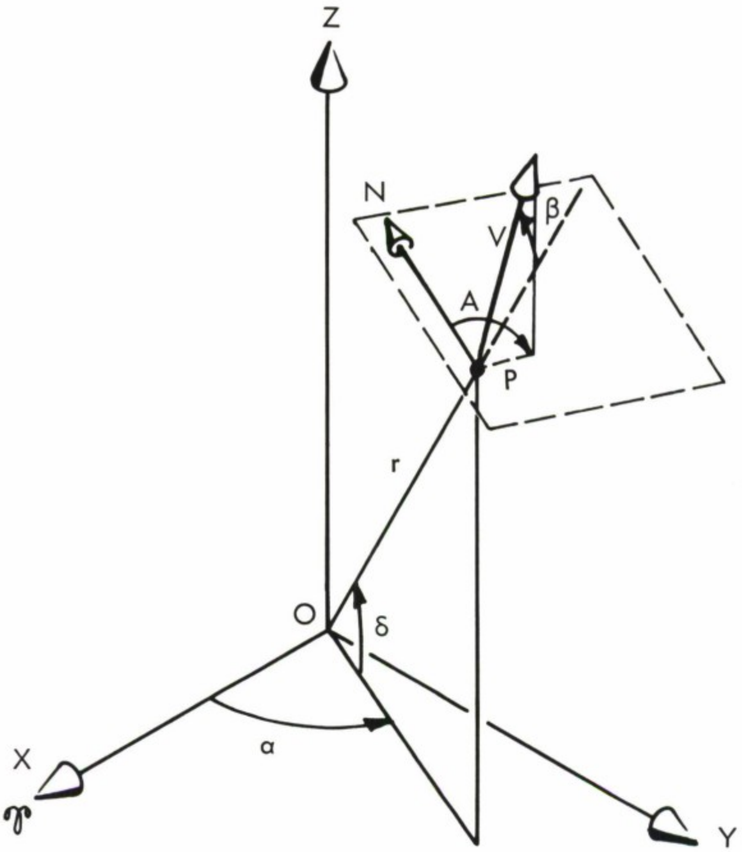
where

O	is the geocenter
V	is the velocity vector
X	is a vector from O in the equatorial plane directed to the true vernal equinox at 0.0h universal time on the day of epoch
Y	is a vector from O and perpendicular to X such that (X, Y, Z) is a right-handed system
Z	is a vector perpendicular to the equatorial plane and directed north.

In $P = P(x, y, z, \dot{x}, \dot{y}, \dot{z})$, x, y, z are components of position of the body in the X, Y, Z directions respectively, and $\dot{x}, \dot{y}, \dot{z}$ are its components of velocity in these directions.

7.2 GEOCENTRIC POLAR SPHERICAL (ADBARV) SYSTEM

The position and velocity of a body at point P are P = P(α, δ, β, A, r, v)



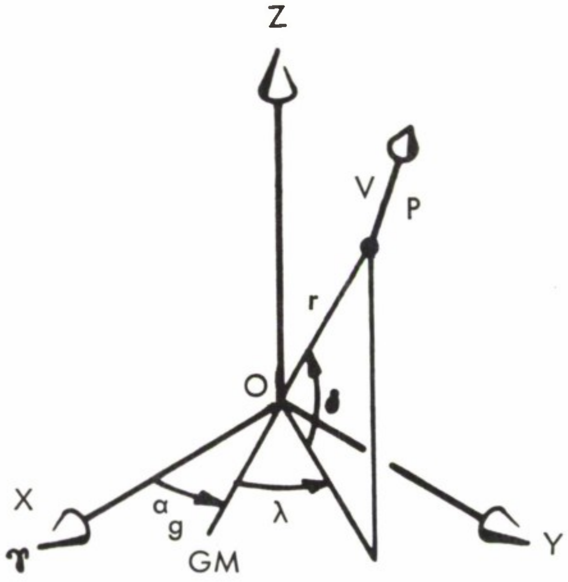
where V is a vector equal in magnitude and direction to the velocity of the body at point P, and where X is a vector from O in the equatorial plane directed to the true vernal equinox at 0.0h universal time on the day of epoch.

In P = P(α, δ, β, A, R, v)

- α is the right ascension of P
- δ is the declination of P
- β is the flight path angle measured positive downward from the geocentric vertical at P to the velocity vector
- A is the azimuth of the velocity vector measured positive clockwise from true north to the projection of the velocity vector in a plane normal to the local geocentric vertical
- r is the geocentric range to P
- v is the magnitude of the velocity vector, V.

7.3 GEOCENTRIC POLAR SPHERICAL (λDBARV) SYSTEM

The position and velocity of a body at point P are $P = P(\lambda, \delta, \beta, A, r, v)$



where

X is a vector from O in the equatorial plane directed to the true vernal equinox at 0.0h universal time on the day of epoch

α_g is the right ascension of the Greenwich meridian at time t

$$\alpha_g = \alpha_{go} + \omega_e (t - t_M)$$

α_{go} is the right ascension of the Greenwich meridian at time t_M

t_M is 0.0h universal time at day of epoch

ω_e is the rate of earth rotation.

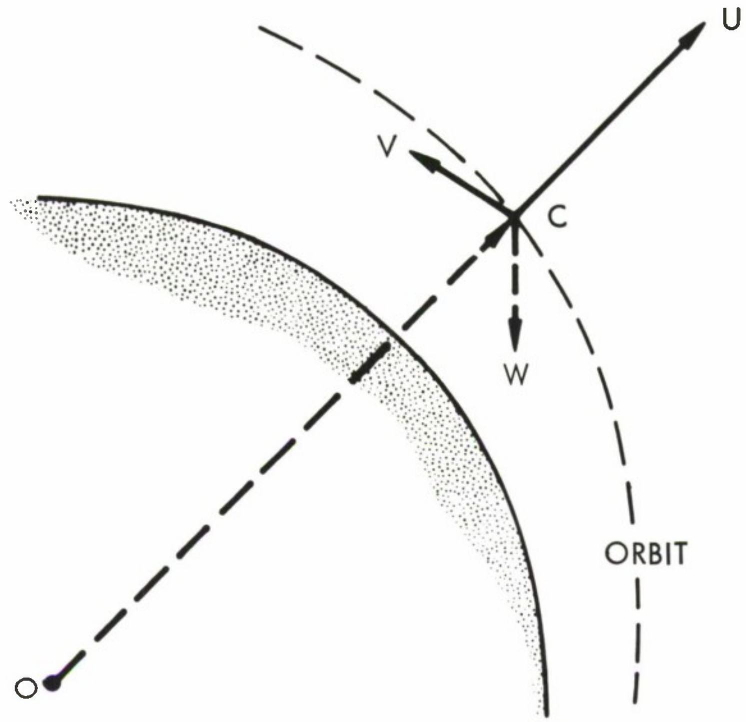
In $P = P(\lambda, \delta, \beta, A, r, v,)$

λ is longitude of P, measured positive eastward from the Greenwich meridian.

δ, β, A, r, V are the same parameters as defined in Section 7.2.

7.4 ORBIT PLANE (U, V, W) SYSTEM

Deviations in position and velocity of a body at C are C = C $(u, v, w, \dot{u}, \dot{v}, \dot{w})$



where

O is the center of the earth

C is the center of a body in orbit

U is the vector from C collinear to a vector from O to C

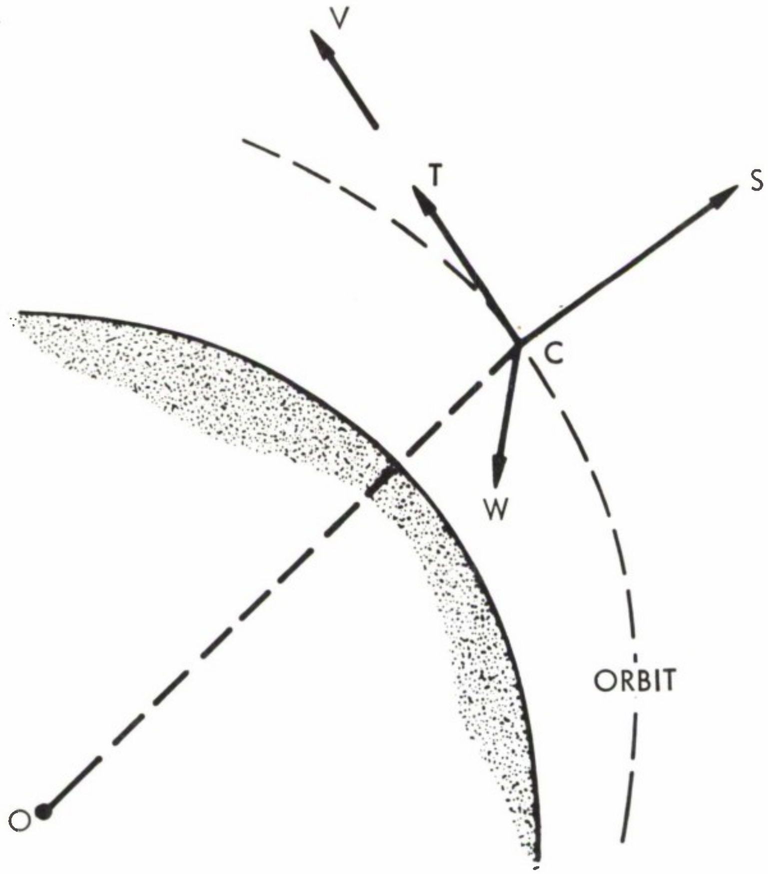
V is the vector from C perpendicular to U and lying in the orbit plane

W is the vector from C which completes a right handed coordinate system

In C = C $(u, v, w, \dot{u}, \dot{v}, \dot{w})$, u, v, w are the components of the deviation in position of the body in the up, down, cross or u, v, w directions respectively; and $\dot{u}, \dot{v}, \dot{w}$ are its components of deviation in velocity in these directions.

7.5 ORBIT PLANE (S, T, W) SYSTEM

Deviations in position and velocity of a body at C are C = C $(s, t, w, \dot{s}, \dot{t}, \dot{w})$



where

O is the center of the earth

C is the center of a body in orbit

T is the vector from C collinear with the instantaneous velocity vector of the body

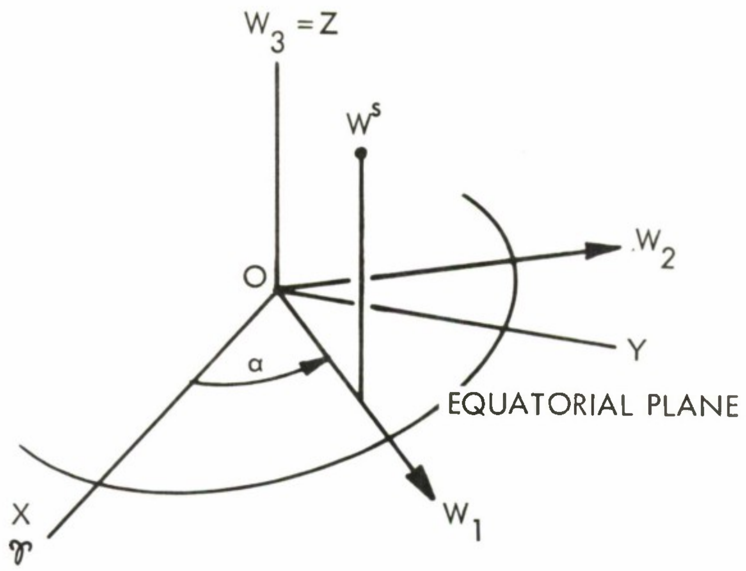
S is the vector from C perpendicular to T and lying in the orbit plane

W is the vector from C which completes a right handed system

In C = C $(s, t, w, \dot{s}, \dot{t}, \dot{w})$ s, t, w, are the components of the deviation in position of the body in the S, T, W, directions respectively; and $\dot{s}, \dot{t}, \dot{w}$ are its components of deviation in velocity in these directions.

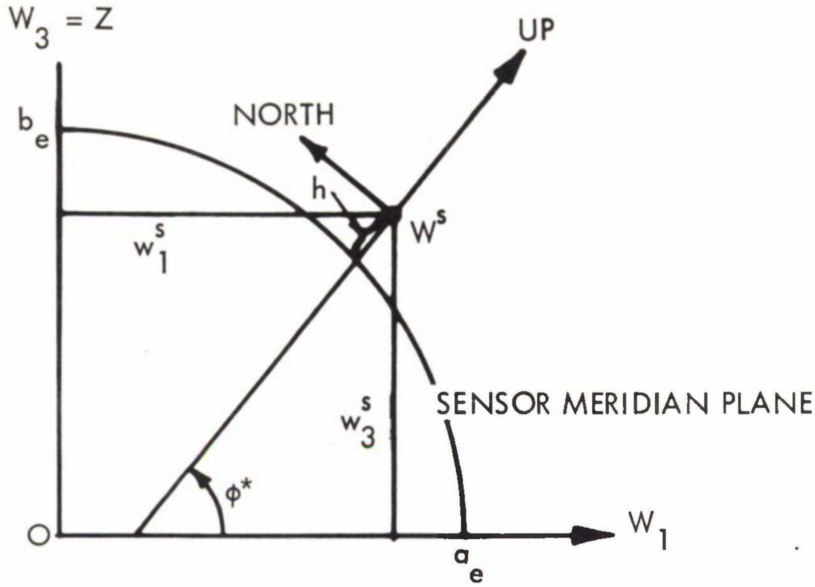
7.6 SENSOR-DEPENDENT (W) COORDINATE SYSTEM

The position of a sensor and an astronomical body relative to the sensor at points W^s, W are described by $W^s = W^s(w_1^s, w_2^s, w_3^s)$ and $W = W(w_1, w_2, w_3)$.



where

X	is a vector from 0 in the equatorial plane directed to the vernal equinox
W^s	is the location of the sensor at some time t
α	$= \lambda + \alpha_{g_o} + \omega_e(t - t_M)$
λ	is the geodetic longitude of the sensor, where X is true of 0.0h day of epoch
α_{g_o}	is the right ascension of Greenwich at time t_M
t_M	is 0.0h universal time of epoch
ω_e	is the rate of earth rotation
W_1, W_2	are the axes X, Y, rotated through the angle α
O	is the geocenter.

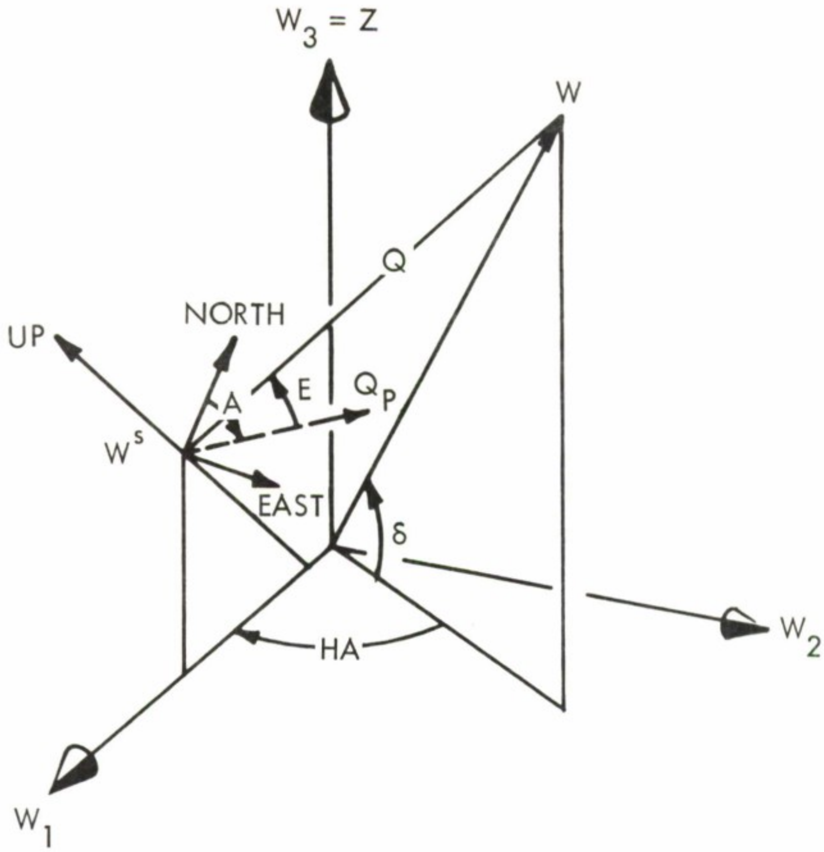


where

a_e	is the semimajor axis of the earth
b_e	is the semiminor axis of the earth
h	is the sensor altitude above the geoid
ϕ^*	is the geodetic latitude, where X and W_1 are with respect to true of date.

In $W^s = W^s(w_1^s, w_2^s, w_3^s)$

w_1^s	is parallel to the equatorial plane and in the sensor meridian plane
w_2^s	is normal to the sensor meridian plane to form a right-handed system, and equals zero since the sensor is in the meridian plane
w_3^s	is perpendicular to the equatorial plane and defines a distance from it in the sensor meridian plane.



where

W	is the position of the body
W^s	is the position of the sensor
Q_p	is the projection of $Q = W - W^s$ onto the tangent plane at W^s
HA	is the hour angle from W to W^s
δ	is the declination of W
A	is azimuth of W from W^s
E	is elevation of W from W^s
Q	is the position vector of W from W^s, and other quantities are as previously defined.

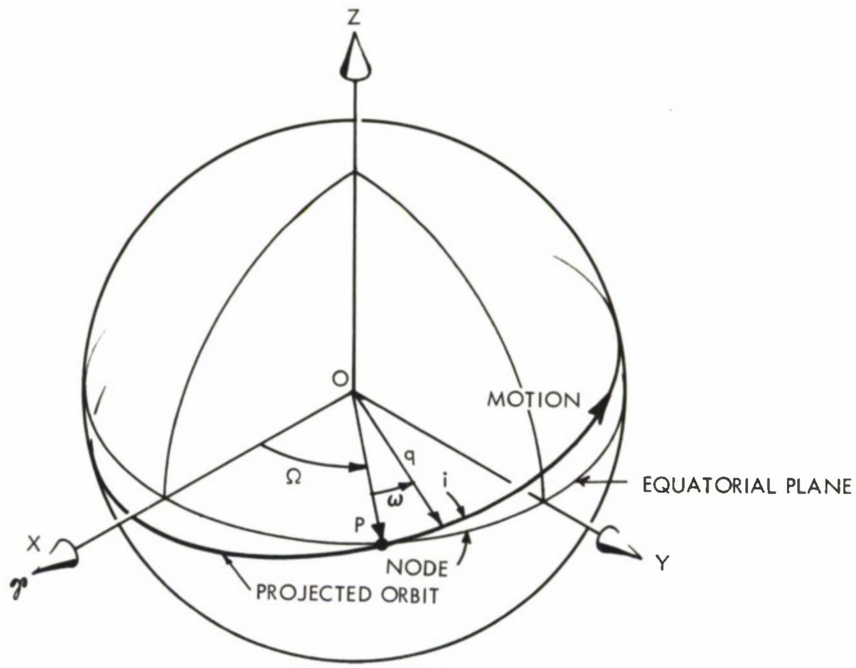
In $W = W(w_1, w_2, w_3)$

w_1 and w_3 are defined the same as w_1^s and w_3^s preceding

w_2 is defined the same as w_2^s except it is not necessarily zero.

7.7 OSCULATING CLASSICAL ELEMENTS

The position and velocity of a body at point $P = P(T_o, P_N, N_o, C_N, \Omega_o, \dot{\Omega}_o, \omega_o, \dot{\omega}_o, i, e, Q_o)$. These elements are defined at the time of the ascending node of the osculating ellipse. These elements are printed only for an update time.



where

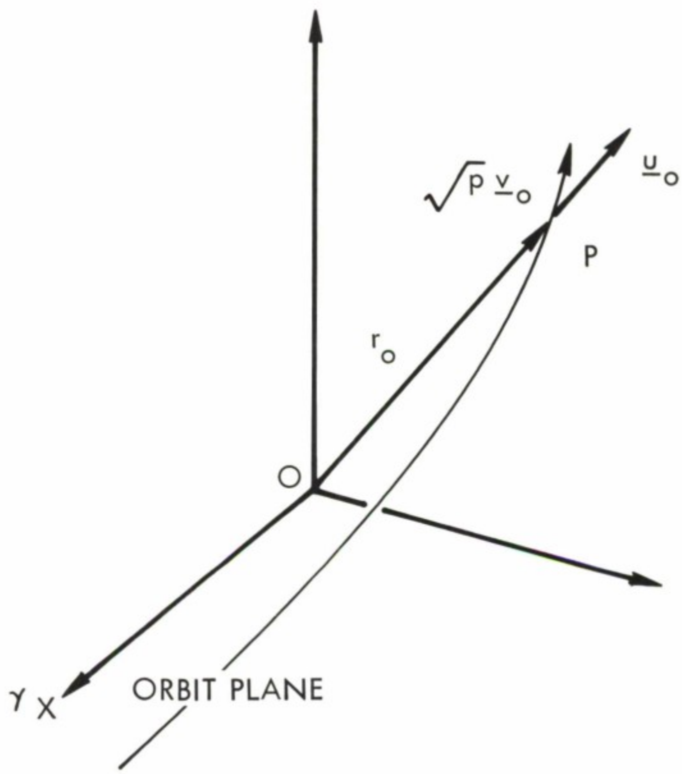
X is a vector from O in the equatorial plane directed to the true vernal equinox at 0.0h universal time on the day of epoch

In $P = P(T_o, P_N, N_o, C_N, \Omega_o, \dot{\Omega}_o, \omega_o, \dot{\omega}_o, i, e, Q_o)$

- T_o is the time of epoch, nodal crossing time for epoch revolution, in days of year
- P_N is the nodal period at epoch
- N_o is the epoch revolution number
- C_N is the rate of change of nodal period (P_N)
- Ω_o is the right ascension of ascending node at T_o
- $\dot{\Omega}_o$ is the time derivative of right ascension of ascending node
- ω_o is the argument of perigee at T_o
- $\dot{\omega}_o$ is the time derivative of argument of perigee
- i is the orbit plane inclination to equatorial plane
- e is eccentricity
- Q_o is the perigee distance at T_o

7.8 INDETERMINACY FREE ELEMENTS

The position and velocity of a body at point P are P($1/a$, r_o, $\underline{u}_o$, $\sqrt{p}$, $\underline{v}_o$, D_o). These are called the indeterminacy free osculating elements at the given update times. They have been included to circumvent the indeterminacies which are inherent in the "classical set" (a, e, Ω, i, ω, T) for certain types of orbits. For example: when $i = 0$, Ω is undefined; when $e = 0$, ω is undefined.



where

X is a vector from O in the equatorial plane directed to the true vernal equinox at 0.0h universal time on the day of epoch.

In $P = P(1/a,\ r_o,\ \underline{u}_o,\ \sqrt{p}\underline{v}_o,\ D_o)$

$1/a$ is the inverse of the semimajor axis

r_o is the magnitude of the position vector at the update time

$\underline{u}_o$ is the unit vector collinear with the position vector at the update time

$\sqrt{p}\underline{v}_o$ is the vector in the orbit plane, orthogonal to $\underline{u}_o$, with magnitude of the square root of the semi-latus rectum

D_o is the scalar product of position and velocity vectors at the reference time

In order that a set of orbital elements be useful, it should provide a description of the orbit that is easily understood, as well as define position and velocity at epoch. Ease of two-body position and velocity prediction is also of importance. The indeterminacy free elements are useful because (a) they are determinate for all types of orbits; (b) they retain some descriptive value which is nearly equal to the "classical set," and certainly better than $\underline{r}_o$ and $\dot{\underline{r}}_o$; and (c) two-body position and velocity predictions are easily accomplished using a single set of equations.

The equations of condition on the unit vectors are as follows:

$$\underline{u}_o \cdot \underline{u}_o = 1$$

$$\sqrt{p}\underline{v}_o \cdot \sqrt{p}\underline{v}_o = p$$

$$\sqrt{p}\underline{v}_o \cdot \underline{u}_o = \underline{u}_o \sqrt{p}\underline{v}_o = 0$$

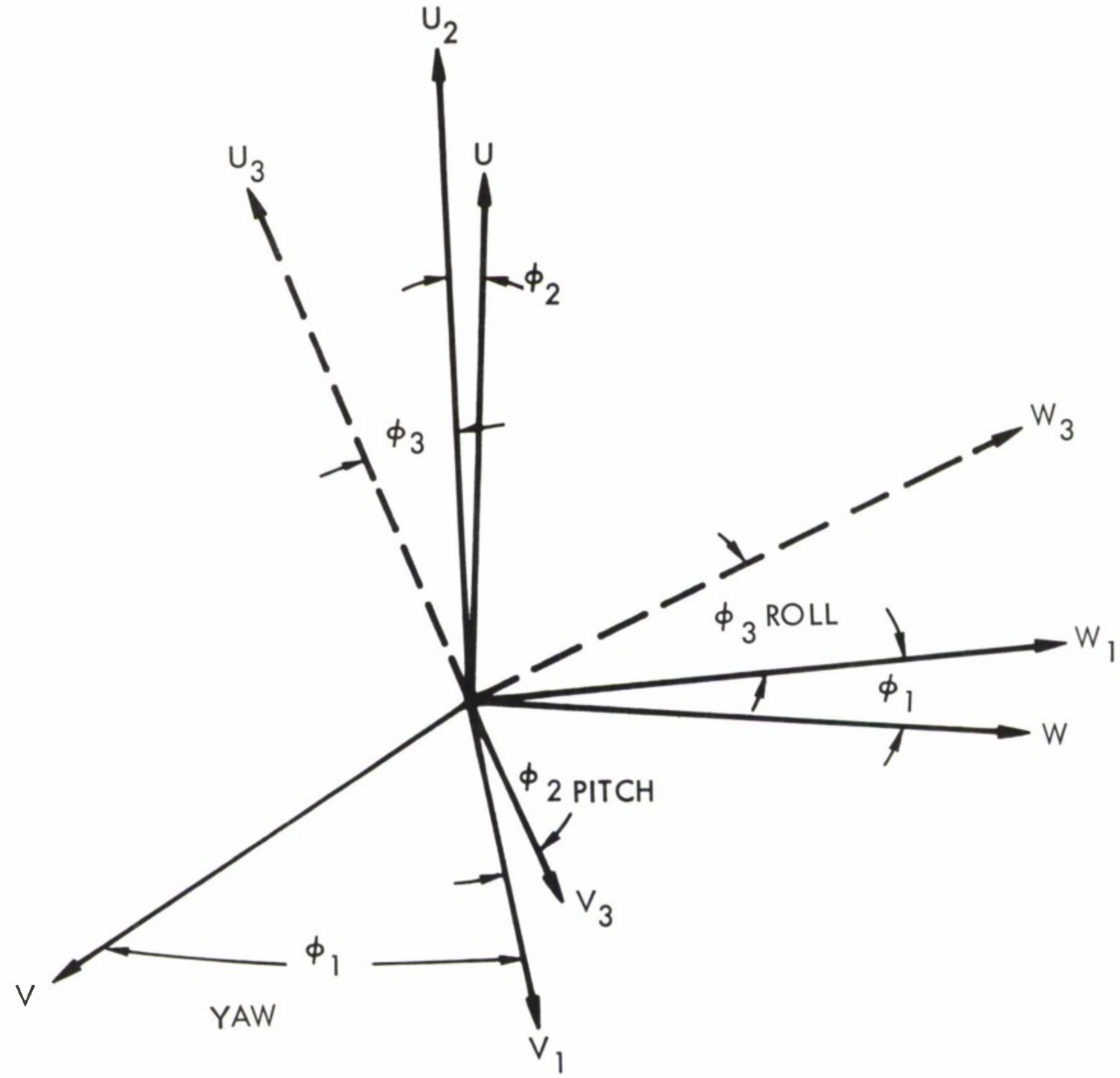
There are six independent orbital elements; i.e., nine elements related by three equations. The manner in which these elements reduce to the minimum set required to define each orbit type is detailed in Table 7-I.

Table 7-I. Summary of Conditions Necessary to Define Each Orbit Type with Indeterminacy Free Elements

Orbit Type	Required Number of Elements	$\frac{1}{a}$	r_o	D_o	$\sqrt{p}\,\underline{v}_o$	$\underline{u}_o$	Total Elements	Equations of Condition
Circle	4	$a = r_o$		0			7	3
Ellipse	6						9	3
Parabola	5	0					8	8
Hyperbola	6						9	3
Rectilinear Ellipse	4			Derived from $\frac{1}{a}$, r_o	0		5	1
Rectilinear Parabola	3	0		Derived from $\frac{1}{a}$, r_o	0		4	1
Rectilinear Hyperbola	4			Derived from $\frac{1}{a}$, r_o	0		5	1

7.9 ERROR ELLIPSOID ROTATION

The orientation of the position error ellipsoid with respect to the U. V, W axes is such that the three principal axes are identified by the nearest axes of the U, V, W set.



Illustrated are the ordered yaw (ϕ, positive right, about axis) pitch (ϕ_2, positive clockwise about V_2 axis) rotations of the U, V, W coordinate system which align it with the error ellipsoid. Note that V is approximately the direction of flight of the vehicle. Yaw is about the U axis; pitch is about the W_1 axis (newly positioned as a result of first rotation, ϕ); and roll is about the V_3 axis (newly positioned as a result of the ϕ_1 and ϕ_2 rotations).

Yaw is positive to the right, thus ϕ_1, as shown, is a negative yaw; pitch down, thus ϕ_2, as shown, is a negative pitch; and roll is positive clockwise when facing along the positive V axis; thus, ϕ_3 as shown, is positive.